PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification :
H04M 11/00, 15/00, 3/42
H04M 7/00

A1

(11) International Publication Number: WO 94/05111

(43) International Publication Date: 3 March 1994 (03.03.94)

(21) International Application Number: PCT/US93/07835

(22) International Filing Date: 20 August 1993 (20.08.93)

(30) Priority data:

934,240	25 August 1992 (25.08.92)	US
972,501	6 November 1992 (06.11.92)	US
972,502	6 November 1992 (06.11.92)	US
972,503	6 November 1992 (06.11.92)	US
972,504	6 November 1992 (06.11.92)	US
972,813	6 November 1992 (06.11.92)	US
972,817	6 November 1992 (06.11.92)	US

(71) Applicant: BELL COMMUNICATIONS RESEARCH, INC. [US/US]; 290 West Mount Pleasant Avenue, Livingston, NJ 07039-2729 (US).

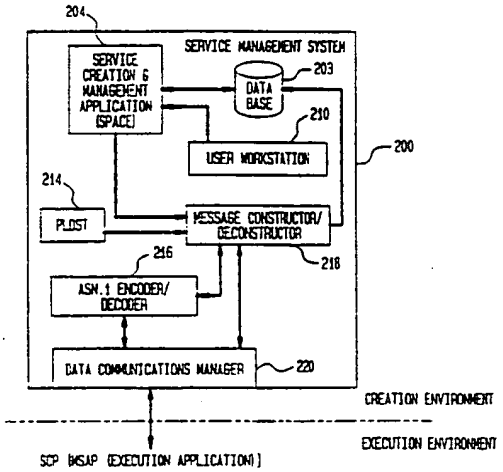
(72) Inventors: MAN, Susan, Kuo-Kung ; 5 Old Farm Road, Holmdel, NJ 07733 (US). CEBULKA, Kathleen, Dickens ; 75 Evans Avenue, Piscataway, NJ 08854 (US). FISHER, Gregory, Michael ; 2704 North Oak Boulevard, North Brunswick, NJ 08902 (US). LANCASTER, Jill, Ellen ; 1029 Old York Road, Hightstown, NJ 08520 (US). NAZIF, Zaher, Amin ; 7 Jane Lane, High Bridge, NJ 08829 (US). VINCIGUERRA, Lori, Jean ; 75 Evans Avenue, Piscataway, NJ 08854 (US). SHASTRY, Subramanya, Kolpe ; 66 Green Meadow Boulevard, Middletown, NJ 07748 (US). CHU, Francis, Yiu-keung ; 123 Forest Glen Drive, Highland Park, NJ 08904 (US). WANG, Lei-Lei, Chu ; 37 Brians Circle, Princeton Junction, NJ 08550 (US).

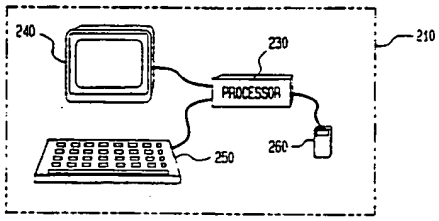
(74) Agents: WINTER, Richard, C.; PCT International, Inc., Post Office Box 573, New Vernon, NJ 07976 (US) et al.

(81) Designated States: CA, JP, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: AN APPARATUS AND METHOD FOR CREATING, TESTING, VALIDATING, AND PROVISIONING TELECOMMUNICATION SERVICES





(57) Abstract

In a method of creating a general service specification for a call processing record in a telephone network, a processor [230] in the record creation system [200] prompts the operator to identify at least one optional node [128c], at least one required node [126c], and at least one restricted node from a node set presented to the operator. Also, in a method of creating a template for the creation of call processing services, a processor [230] in the record creation system [200] displays a selected call processing record [925] to the operator. The operator then identifies which nodes in the call processing record will be made customizable. Data tables [1220] can be created and accessed by one or more call processing records for executing telephone services. Also, call processing sample nodes [734] and measurement nodes [733] can be created and used for call processing.

AN APPARATUS AND METHOD FOR CREATING, TESTING, VALIDATING, AND PROVISIONING TELECOMMUNICATION SERVICES

Background of the Invention

The present invention relates generally to the field of customized services, and more specifically to the problems of creating, testing, validating, and provisioning customized telecommunication services.

Existing telephone systems can include a service creation environment for creating customized telephone services and a service execution environment for executing the telephone services. The service creation environment can include a graphical user interface, which permits a user to build and/or change a displayed graphical representation of a desired service using "nodes," "decision boxes," and "branches." Each node represents a high level instruction for the execution of the service. The displayed graphical representation of the service is translated to a binary representation and stored as a call processing record (CPR). CPRs are transmitted from a creation environment to an execution environment where they are executed during call processing operations to send call processing instructions to inquiring switches.

These systems and methods for creating and executing customized telephone services can be implemented in the Advanced Intelligent Telephone Network (AIN).

Fig. 1 illustrates an exemplary AIN comprising System Service Points (SSPs) 30, 35, 40, and 45, Signal Transfer Points (STPs) 48 and 50, Service Control Points (SCPs) 10 and 20, and Service Management Systems (SMS) 60 (only one shown). SSPs are central office switching systems which receive telephone calls from telephones 12. Each SSP recognizes a variety of "triggers" within customer telephone call signals and generates queries to SCPs based on the triggers. The SSPs then process customer calls in response to commands received from the SCPs.

The SCPs communicate with the SSPs over a common-channel-signalling (CCS) network 67 that includes STPs 48 and

each customer with substantially the same service. Hence, it would be beneficial to an operating company to be able to provide a specification for a service from which numerous similar graphs could be generated, but with enough flexibility to cater to each customer's individual needs. For example, an operating company may determine that many of its customers are interested in a service that permits the customer to specify the carrier for long distance calls associated with the customer's "800" number. This service would be similar for each customer and would require certain nodes (such as carrier nodes) in each customer's CPR. However, larger businesses may want additional features from such a service. For example, they may want to provide for different carriers during different times of the day. It would therefore be beneficial to the operating company to be able to offer a basic 800 service and an enhanced 800 service wherein each service is partially predefined, yet flexible enough to permit some customization by the individual customers.

Accordingly, it is desirable to provide a general service specification that allows a service creator to define a service, but permits a user enough flexibility to customize the service to some degree.

It is also desirable to permit a service creator to define a service specification in which certain predetermined nodes are mandatory, certain predetermined nodes are optional, and certain predetermined nodes are restricted.

In addition, many customers may want the same service, or they may want services with only minor differences. For example, an operating company may determine that most of its customers desire a service that permits them to specify the carrier for their long distance calls. This service would be similar for each customer, and each customer's graph for this service would be almost identical. It may be impractical or costly for the service creator to generate essentially the same CPR numerous times, once for each customer, particularly when only slight differences need exist in the CPRs. In the

obtain useful information concerning the execution of the service. Accordingly, it is desirable to permit a service provider to administer the execution of a service.

In the system referred to, services can be created using only fixed or predefined nodes. Although these nodes provide a great deal of service creation flexibility, because only certain nodes are available, service creation flexibility is limited.

It is desirable to provide for the design, layout, and instantiation of user-defined nodes that are indistinguishable from other predefined nodes from the perspective of the service creation and execution environments.

The CPRs discussed above comprise a "key" and a plurality of nodes, decision boxes, and branches. The "key" includes a telephone number and a suffix. The suffix .e04 means that the CPR controls calls made from the corresponding telephone number, and the .e05 suffix means that the CPR controls calls made to the corresponding telephone number. Hence, to provide separate services for calls made to or from a subscriber's telephone number, existing service creation systems require separate CPRs.

Requiring multiple CPRs per customer in a system having many customers strains the storage and execution environments with tremendous amounts of service logic. Moreover, it complicates and hinders efficient service execution and management.

Accordingly, it is desirable to provide a CPR structure that permits efficient use of CPRs on a large scale in an execution environment.

It is also desirable to provide a CPR structure that permits quick and efficient storage, access, management, and execution of CPRs.

Additional desires of the invention will be set forth in the description which follows, and in part will be apparent from the description, or may be learned by practice of the

predetermined expressions; displaying all expressions of the selected node; prompting the operator to specify which of the selected node expressions will be customizable; designating the specified node expressions as customizable; and enabling the selected call processing record and the designation of customizable node expressions for the selected node as a service template.

To achieve the foregoing desires and objects, and in accordance with the purposes of the invention as embodied and broadly described herein, the present invention also provides in a telecommunication service creation environment providing for call processing records and value tables, the value tables comprising one or more columns and one or more rows of values, a method of creating a call processing procedure to determine whether a particular value exists in a particular value table, the method comprises the steps, executed by a data processor, of: prompting an operator to name a value table to be searched; receiving from the operator a name of the value table to be searched; prompting an operator to identify one or more columns in the value table to be searched; receiving from the operator an identification of one or more values in the value table to be searched; prompting an operator to specify a value to be searched for in the one or more columns to be searched; receiving from the operator a value to be searched for in the one or more columns to be searched; prompting an operator to specify comparison criteria for the value specified and values in the column to be searched; receiving from the operator a comparison criteria for the value specified and values in the column to be searched; and instantiating the table name, one or more columns, value to be searched for, and comparison criteria as part of the call processing procedure.

To achieve the foregoing desires and objects, and in accordance with the purposes of the invention as embodied and broadly described herein, the present invention also provides a method of providing a call processing sample node to determine an amount of call processing activity, the method

procedures selected by the operator; and enabling the underlying representation of call processing procedures as a single node for use in creating call processing records.

Finally, to achieve the foregoing desires and objects, and in accordance with the purposes of the invention as embodied and broadly described herein, the present invention provides a call processing record for execution in a telephone service execution environment, comprising: a record header associating the call processing record with a corresponding telephone service subscriber; at least one call processing logic section including call processing procedures executable by a processor in the telephone service execution environment; at least one first data section, each of the at least one first data sections being associated with one of the at least one call processing logic sections and storing data executable only by the call processing procedures included in the associated one of the at least one call processing sections; and at least one entry point, each of the at least one entry points being associated with one of the at least one call processing logic sections and an associated one of the at least one first data sections, the at least one entry point identifying the associated one of the at least one call processing sections.

Brief Description of the Drawings

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate presently preferred implementations of this invention and, together with the general description given above and the detailed description of the preferred implementations given below, serve to explain the principles of the invention.

In the drawings:

Fig. 1 is a block diagram of the Advanced Intelligent Telephone Network (AIN);

Fig. 11 is a flow diagram illustrating a GSS creation operation in accordance with one embodiment of the present invention;

Fig. 12 illustrates a GSS Editor screen showing an exemplary GSS in accordance with one embodiment of the present invention;

Fig. 13A illustrates an example of a graph in accordance with one embodiment of the present invention;

Fig. 13B illustrates another example of a graph in accordance with one embodiment of the present invention;

Fig. 14 is a flow diagram illustrating an operation for validating a graph against an associated GSS in accordance with one embodiment of the present invention;

Fig. 15 illustrates a NODE Editor screen in accordance with one embodiment of the present invention;

Fig. 16 illustrates an example of a graph using Measurement and Sampling nodes in accordance with one embodiment of the present invention;

Fig. 17 illustrates an example of a graph using External System Interaction nodes in accordance with one embodiment of the present invention;

Fig. 18 illustrates a Custom Node Editor screen in accordance with one embodiment of the present invention;

Fig. 19A illustrates Parameter Editor screen in accordance with one embodiment of the present invention;

Fig. 19B illustrates a Selection List Editor screen in accordance with one embodiment of the present invention;

Fig. 20 illustrates a Custom Node Preview screen in accordance with one embodiment of the present invention;

Fig. 21 illustrates a Custom Node Layout Screen in accordance with one embodiment of the present invention;

Fig. 22 illustrates a Custom Node Category screen in accordance with one embodiment of the present invention;

Fig. 23 illustrates an example of a graph using an Intable node in accordance with one embodiment of the present invention;

A. System Configuration

In a preferred embodiment of the present invention, a service is created in the AIN. In particular, a service is created by a user at a workstation associated with the SMS 200.

Fig. 2A is a block diagram of a preferred embodiment of an SMS 200 in accordance with the present invention. The SMS 200 includes a service creation and management application 204 which preferably comprises the SPACE® application version 2.0. SPACE is a proprietary software application owned by Bellcore, the assignee of this application.

In addition to the service creation and management application 204, SMS 200 includes a user workstation 210. Preferably, user workstation 210 (also shown in Fig. 2B) includes an IBM RS-600 (Model 320) as well as related equipment, for example, processor 230, keyboard 250, mouse 260, and graphical display 240 which preferably runs AIX windows (IBM), version 3.2 or X-windows, version 11, release 4 or later .

The SMS 200 also includes database 203, Programming Language Data Structure Translator (PLDST) 214, ASN.1 Encoder/Decoder 216, Message Constructor/Deconstructor (Message C/D) 218, and Data Communications Manager 220. These elements, their relationships, and their relationship to the execution environment in an SCP 10, 20 are described in the incorporated interface application.

The service creation portion of the SPACE application is dedicated to the creation of CPRs and Tables (described below). CPRs are created using the SPACE application by generating a high level, displayed representation (graph) of the desired service on the display 240 of user workstation 210. The displayed graph of a CPR is extremely useful in that it permits an operator to create and understand the telephone service being created and to test and validate the service logic. However, the graph cannot be interpreted

1. Display and Editing Modules

In a preferred embodiment, the display procedures 300 of Fig. 3 include display and editing modules. The display and editing modules display various graphical objects on the display 240 of workstation 210 and allow manipulation of the graphical objects by the user. The display and editing modules, as shown in Fig. 4A, include Record Control module 321, Node Specification Editing module 322, CPR Editing module 323, GSS Editing module 324, Graph Editing module 325, Variable Editing module 326, Form Creation module 327, Provisioning module 328, Table Node Editing module 339, and Dialog module 329.

Record Control module 321 interfaces Database module 340 (Fig. 4C) with each of the editing modules (modules 322, 323, 324, 325, 326, and 339) to transfer data from database 203 to editor buffers (not shown) associated with the respective editing modules in the workstation 210 and to transfer (save) data from the editor buffers to database 203. Record Control module 321 also allows a user to prepare a template (described below in section G) for a mass market service.

CPR Editing module 323 allows a user to change the characteristics (i.e., headers, entry points, etc., as described below) of a CPR. To do so, CPR Editing module 323 invokes the Graph Editing module 324 and the Variable Editing module 326 to change corresponding portions of the CPR. The CPR Editing module 323 also allows editing of existing templates.

Graph Editing module 325 allows a user to manipulate the structure or relationship of nodes and branches in a graph. Thus, in conjunction with the Node Specification Editing module 322 and Variable Editing module 326, which allows manipulation of call variables within nodes, the Graph Editing module 325 also allows graphs to be edited and translates the corresponding internal data structures into graphical display representations for display on the display 240 of workstation 210. In addition, the Graph Editing

Telephone Numbering Plan, I = International Number, S = Special Number, and P = Private Number.

d. String - This data type is a string of characters.

e. Numeric String - This data type is a string of digits, "#," or "*," as can be entered from a telephone keypad.

f. Date - This data type represents a date.

g. Day of Week (DOW) - This data type is used to represent the days of the week.

h. Time of Day (TOD) - This data type is used to represent the time of day.

i. Carrier - The Carrier data type is used to represent an Inter- or Intra-LATA Telephone Carrier Company Designation. For example, LEC, ATX, or 222.

j. Boolean - This data type is used to represent one of only two possible values such as true/false or yes/no.

k. Float - This data type is used to represent a floating point number. The precision is determined by storage restrictions.

l. Signaling Point Code - This data type represents information about network signaling.

m. Measurement Vector - This data type represents a vector of counters.

n. Table - This data type is a table of rows and columns where data is stored (see Section C.2 below).

The Variable Editing module 326 is also used to restrict input values, identify data for templates, and specify user prompt language. In addition, the Variable Editing Module 326 is used to define user input parameters when creating User Defined Nodes (described below in Section F.5).

General Service Specification (GSS) Editing module 324 is used to retrieve, display, and edit a GSS (described below in section E).

Node Specification Editing module 322 allows a user to change the characteristics of a node specification, and thereby define a custom or User-defined node. This module

these data structure modules is particularly related to one or more data structure types.

Upon creation of a graph, the Graph module 331 is invoked to define the data structure which results upon creation of the logical relation between branches and nodes in the graph. Within the Graph module 331, data structures representing individual branches within the graph are further defined by the Branch module 333. Thus, at points in the graph where a branch is required, the Graph module 331 invokes the Branch module 333. Data structures representing individual nodes within the graph are further defined by the Node module 332. Thus, at points in the graph where a node is required, the Graph module 331 invokes the Node module 332. Similarly, expressions within a node are defined by the Expression module 334, which is called as necessary by the Node Specification Editing module 332.

As previously described, preferred implementations of the present invention use object oriented-programming techniques. One aspect of object oriented-programming is that all functions operable upon a particular "object" are defined with the object. Thus, all functions operable upon a graph ("the object") are defined within the Graph module 331. Accordingly, each data structure module preferably represents the data structure (i.e., defines the structure) and allows manipulation (i.e., defines the operable functions) of that data structure. Data structure modules may also use subordinate data structure modules as described above.

CPR module 330 internally represents and allows manipulation of graphs and call variables which define a customer service. This module also handles the representation and manipulation of templates. The CPR module also includes administrative information such as, for example, record ownership and status information. The CPR module 330 invokes Graph module 331 and Variable module 336.

Call Variable module 336 represents and handles manipulation of different types of call variables used in graphs and data sections of CPRs. This module reads a set of variable expressions from a series of files in the database 203. A preferred implementation provides for two types of variables: call variables used in CPRs and node specification parameters used in user-defined nodes.

Generic Service Specification (GSS) module 337 represents and handles manipulation of objects which specify the type of service a graph may represent.

3. Database and Related Processing Modules

As shown in Fig. 4C, the binary procedures 308 in Fig. 3 preferably include Database module 340, Binary module 341, Validation module 342, and Testing module 343. Binary module 341 converts various internal data structures into binary representations that can be transferred between different hardware configurations. This module also performs the reverse process of converting binary representations of CPRs and tables into internal data structures.

Database module 340 stores, retrieves, deletes, and searches on CPRs, templates, user-defined nodes, GSSs, and tables in database 203.

Validation module 342 facilitates CPR validation procedures.

Finally, Testing module 343 simulates call processing execution and produces a resulting "processed" binary representation.

C. System Records

The foregoing hardware and software components cooperate to allow a user to create customer services. Preferably, services are created by the formation of two types of system records: CPRs and tables.

global data may include, for example, declarations and/or definitions of call variables, embedded tables, and measurement vectors.

c. Entry Points

An entry point in a CPR is a point at which call processing can be initiated. Each entry point corresponds to a previously defined graph and an associated local data section, the interpretation and execution of which establishes a customer service. As seen from Fig. 5, a CPR may have more than one entry point; hence, all of a customer's services may be provided on a single CPR.

A user may assign any name to an entry point. Entry points are preferably grouped as "trigger" and "non-trigger" entry points. For example, two entry points have special significance in the execution environment: (1) "ani" which is called to process an originating number query; and (2) "dln" which is called to process a called number query.

Non-trigger type entry points are preferably used by other entry points within the CPR or other CPRs.

d. Local Data Sections

As shown in Fig. 5, each entry point 406 is associated with a local data section 408. The local data section 408 includes local data used only by the corresponding logic section of the associated entry point. This local data includes definitions of call variables of local scope.

e. Logic Sections

Logic section 410 contains the actual call processing logic or call processing procedure corresponding to a particular graph or service.

When a SCP 202 processes a CPR in the execution environment, after having retrieved the CPR based on the CPR

NUMBER." The data type of the EXTENSION column is a numeric string, and the data type of the TELEPHONE NUMBER column is a telephone data type. The maximum length of the numeric string in the EXTENSION column is four digits, and the maximum length of the TELEPHONE NUMBER in the telephone column is 15 digits. The key specification 514 permits a user to specify which column uniquely identifies a row and allows for more efficient search.

Fig. 6C illustrates a table record structure 518 for a stand alone table. As shown, the structure includes a header section 516, the table specification 506 as shown in Fig. 6B, and the table data 500 as shown in Fig. 6A. For embedded tables, the table specification 506 and table data 500 are stored as part of the call variable that denotes the embedded table.

In a preferred implementation, six operations can be performed on table data: addRow, delRow, updtRow, findRow, selRow, and nextRow. These operations are executed using menu buttons (not shown) which are displayed in a Table Editor Screen (not shown) that is displayed when a user selects the Table Suboption 175d as shown in Fig. 7. The addRow operation adds (or inserts) a set of rows into a table. The delRow operation deletes a set of rows in a table. The updtRow operation updates a set of values in a table. The findRow operation searches a table for a specified row. The selRow operation selects a set of column values from a row of a table that matches a specified condition and returns the values from the first row found. The nextRow operation selects a set of column values from the next row of a table that match the specified condition in a previous selRow operation.

D. CPR Creation

A user creates a CPR by accessing a CPR Editor screen on display 240 of workstation 210. To call up the CPR Editor screen, a user logs onto the system (hereafter "system" is

optionally used to indicate whether the service being created has additional forms in other operations systems. The Service Rep field 186 is optionally used to maintain a record of a representative who may have taken a customer's order for the service being created. The New Record Information Dialog Box 180 also includes Controls DTMF Update field 187, which is used to indicate whether the service being created will be used to control the updating of other services or tables.

Once the respective fields in the New Record Information Dialog Box 180 have been filled-in and checked by the user, the user selects the "OK" button, and the system presents the CPR Editor screen 171, as shown, for example, in Fig. 9.

CPR Editor Screen 171 includes a Graph Window Screen 173, a CPR Information window 176, a Graphs In CPR window 178, a Nodes window 179, a Graph Manipulator window 188, a Provisioning Data window 189, Call Variables field 190, and an Entry Point Information dialog box 195.

The user specifies an initial entry point for the CPR using the Entry Point Information dialog box 195. The Entry Point Information dialog box 195 contains two text entry fields: Name field 195a and GSS field 195b. Preferably, a user enters the name of a trigger type entry point (e.g., "ani" or "dln") or a non-trigger type entry point into the Name field 195a. The GSS field 195b is preferably prepopulated with a "generic" GSS, which is a system supplied GSS that includes every node as optional. The user can optionally specify any enabled GSS in the GSS field 195b.

As shown in Fig. 9, some of the information entered in the New Record Information Dialog Box 180 is displayed in the CPR Information window 176 on the CPR Editor screen 171 (i.e., the Type 176a and the Name 176b). The CPR Information window 176 may also include a user's identification field 176c, a modification date(s) field 176d, and an activation or effective date field 176e for the CPR.

The Graphs In CPR window 178 includes "Add Graph" button 178a, "Delete Graph" button 178b, "Edit Graph" button 178c, "Browse Graph" button 178d, and Graph List 178e.

(button 192c), and "Other" nodes (button 192d). Assignment and Decision nodes are described below in Section F. PAGD nodes do just what their name suggests; during call processing, they play an announcement to the caller, prompting the caller to input information, and collect the information. Based on the node type button 192 selected by the user, the system displays the available node choices corresponding to that node type in Nodes List window 191.

The nodes of a graph are arranged in the Graph window 173 using the node function buttons presented in Node Function window 193. Preferable function buttons include "Change Value" button 193a for changing the value of a node, "Delete Item" 193b for deleting a node or branch from a graph, "Delete Subtree" button 193c for deleting a portion (subtree) of a graph, "Add Branches" button 193d for adding branches to a node, "Connect" button 193e for logically relating two nodes in a graph, and "Hide Subtree" button 193f for removing a graph portion from the CPR Editor screen in order to facilitate graph creation or editing.

The nodes of a graph are manipulated in the Graph window 173 using the graph function buttons presented in the Graph Manipulation window 188. Preferable function buttons include "Undo" button 188a for successively undoing graph actions, "Cut" button 188b for removing a subtree from a graph and placing it in an internal buffer, "Copy" button 188d for copying a subtree from a graph and placing in an internal buffer, and "Paste" button 188c for copying a subtree from the internal buffer and placing it in a graph.

Call variables of nodes in a graph are preferably defined using the Call Variables window 190. A user assigns a name to each call variable at "Name" field 190a, the data type of a call variable at the "Data type" field 190b, and the "Value" of a call variable at Value field 190c. The CALL VARIABLE window 190 also includes "Defined In" field 190d to identify the CPR, graph, or node in which the call variable is defined. The "Availability" field 190e defines the scope

A GSS contains information that specifies and describes a generic customer service.

1. GSS Creation

To create a GSS, a user accesses the system screen 170 and selects the "Record" option from menu line 172. When the Record option menu 174 is presented, the user selects the "New" option, and the "New" option suboptions window 175 is displayed. The user then selects the "GSS" suboption 175b. Upon selecting the GSS suboption 175b, a dialog box (not shown) is presented to the user. The dialog box simply requests the user to input a name for the GSS.

After the user inputs a name, the system presents the GSS editor screen 120, as shown, for example, in Fig. 10.

The GSS editor screen 120 preferably includes four sections: GSS Information window 122, GSS Description window 124, Required Nodes window 126, and Optional Nodes window 128. The GSS Information window 122 includes a Name field 122a for the name of the GSS entered by the user, a Creator field 122b for the name of the creator of the GSS, a Modified field 122c for dates on which the GSS has been modified, and an Enable field 122d for a date on which the GSS was enabled.

The GSS Description window 124 is used to enter information regarding the customer service related to the GSS. For instance, the GSS description might contain a detailed description of the service to which the GSS is related or an explanation of the reasons why certain nodes are required, optional, or prohibited within CPRs associated with the GSS. For the "900 Block" service described above, a user may provide the following description: "900 Block is a service directed to residential customers who wish to prevent calls beginning with a 900 area code from their home phones."

A user defines which functions are mandatory or optional within each CPR associated with the GSS by identifying (or listing) required nodes and optional nodes for the GSS in the Required Nodes window 126 and the Optional Node window 128,

Before a subsequent CPR may be associated with the GSS, the GSS must be enabled. To enable a GSS, a user selects the "Operation" option from the first menu line 172 and selects an "Enable" option (not shown) from the Operations options menu (not shown). Preferably, an enabled GSS may not be edited or deleted if other records depend on it, since changes to an enabled GSS could affect records previously associated therewith.

The foregoing description of a method for creating a GSS is summarized in the flowchart shown in Fig. 11. In Fig. 11, a user begins by naming the GSS (step 1000) and describing the GSS and the related service (step 1002). Next, the user defines at least one required node (step 1004), lists the at least one required node (step 1006), defines at least one optional node (step 1008), and lists the at least one optional node (step 1010). Finally, the user stores the GSS in the database (step 1012), enables the GSS (step 1014), and the creation procedure ends (step 1016). In an alternative embodiment, the step of defining at least one restricted node (not shown) would be added. In an alternative embodiment, the user may specify that the GSs has zero or more optional, required, or restricted nodes.

In like manner as described above, a GSS may be created for a template.

2. Validating a CPR in Accordance with an Associated GSS

In accordance with the embodiment of the invention, during a validation process, a graph is examined to determine whether the graph is consistent with the requirements of the associated GSS. If the CPR contains restricted nodes, which are not permitted by the GSS, or does not include the mandatory nodes, the CPR fails the validation process.

Fig. 12 is an example of a GSS Editor screen 120 containing a definition of a GSS named "800basic" for a service that designates a particular long distance carrier

A preferred method by which the present invention validates a CPR graph against its associated GSS is shown, for example, in Fig. 14. In Fig. 14, the system reads the first node in the graph (step 1052) and determines whether the node is a required node (step 1054). If the node is a required node, the system determines whether the node is the last node in the graph (step 1062). If the node is not the last node in the graph, the system goes to the next node in the graph (step 1064) and repeats the procedure. However, if the first node is not a required node, the system determines whether the node is an optional node (step 1056).

If the node is an optional node, the system repeats steps 1062 and 1064. If the node is not an optional node, the node violates the GSS and fails validation (step 1058). This failed validation is displayed to the user (step 1060).

After the final node in a graph is determined (step 1062), the system determines whether every required node of the GSS is present in the graph (step 1050). If not, the graph fails validation. If, however, every required node of the GSS is present in the graph; the system indicates a successful validation to the user (step 1063).

F. Nodes

As discussed in the set of incorporated patent applications, nodes are the basic units that define the logical operations to be performed during call processing. Each node is therefore a separate call processing procedure or a subprocedure of a graph. Nodes are logically connected to form a directed graph.

1. Action Nodes

Action nodes may be categorized as Assignment nodes, Network Action nodes, and Control nodes.

Assignment nodes are nodes which provide a function that sets a designated call variable to a particular value. The

assign "RPBILLNBR = 7033085555." With this assignment, services provided by the CPR having the graph containing the foregoing billingNum node will be billed to telephone number 703-308-5555.

The BillingType node allows a user to assign a value to one or more predefined "billing type" call variables. For example, a billing type call variable may be named RPMONTHLY, may be of signed integer data type, and may have a corresponding expression. Thus, a graph having the foregoing BillingType node allows a user to assign "RPMONTHLY = 15." With this assignment, services provided by the CPR having the graph containing the foregoing BillingType node will be calculated and billed on the fifteenth day of every month.

Control Nodes allow multiple CPR entry points to be traversed as part of a single call execution and include a Handover node and Transfer Control node. The Handover node allows a CPR to call and execute another graph before continuing with the current CPR graph. The graph may be located in another CPR, thus the Handover node requires that the CPR key, trigger, and entry point for the graph be specified within the Handover node. Once the other graph is processed, processing returns to the original CPR graph.

The Transfer Control node is like the Handover node in that another CPR is specified and executed. Unlike the Handover node, however, the processing does not return to the original graph, but remains at the transferred CPR.

2. Decision Nodes

Decision nodes are used to branch execution through the graph. Decisions as to which graph branch to traverse may be made on the basis of a call variable value and an expression within the decision node. For example, a Call Variable Decision node may include a call variable named "READY" of data type Boolean. This decision node branches one way or the other in a graph based on "READY = yes," or "READY = no." Compare nodes compare expressions. For example, a compare

is then superimposed on the CPR editor screen 170. For a sample node, the Node Editor Dialog Box 750 requests a definition of a sampling rate (0-100%) (field 752), collection type (field 753), sampling type (attempted or completed) (field 754), sample node name (field 755), and the call variable to be sampled (field 756). Once the fields are completed and the user selects the "OK" button, the Sample node is instantiated in the graph. Use of a Sampling node in a graph is illustrated in Fig. 16 and described in more detail below.

b. Measurement Nodes

Measurement nodes count events. Events may be, for example, the number of times a graph or a portion of a graph is traversed, how many times a call variable is changed, etc. Measurement nodes may count up or down from a predetermined starting number. Thus, Measurement nodes are used to update a component of a measurement vector. A measurement vector is an "up count" or a "down count."

Measurement nodes are created during graph building by specifying which component of a measurement vector call variable is to be incremented or decremented. This designation is preferably made in the Call Variable window 190 of the CPR Editor Screen 170 (Fig. 9). Alternatively, the measurement vector call variable, the measurement vector component, and the increment/decrement designation are provided in response to prompts in a measurement node Editor Dialog Box (not shown) similar to the Sample Mode Editor Dialog Box 750 shown in Fig. 15.

The system uses a unique counter created when the measurement vector was defined. The counter is loaded with the starting point value and changes the value (up or down) on the basis of subsequent measurements.

Fig. 16 shows part of a graph incorporating a Sample node and Measurement nodes. In this graph, calls originating from a customer's number "3014447500" (header 720) are routed

a. Network Interaction Nodes

Network Interaction nodes preferably include a Connect node, a Terminate node, and a Play Announcement and Collect Digits Node. The Connect node allows a user to route a call to a designated number. The routing number is specified as a call in the Connect node. The Terminate node allows a user to block a call. Once a graph reaches a Terminate node, all call processing is halted. The Play Announcement and Collect Digits node, as discussed above, is used to play an announcement to the customer, and then collect digits (i.e., DTMF signals) from the user in response to the announcement.

b. External System Interaction Nodes

This node type preferably includes a GetData node, SendData node, and WaitForEvent node. The GetData node allows the user to send a message to an external system (outside the SCP) requesting certain data from that external systems data base to be placed in call variables that are specified in the node. The SendData node allows a user to send a message to an external system (outside the SCP) to store certain data as provided in call variables that are specified in the node, in the external system's data base. The WaitForEvent node allows the user to wait for the completion of an external event such as any GetData or SendData operation before call processing will continue.

Fig. 17 illustrates a graph using GetData, SendData, and WaitForEvent nodes. In the graph of Fig. 17, GetData node 1800a requires the SPACE system to get a value from a different system, return it to the SPACE system and put it into a call variable entitled Event 1. Call variable decision node 1800b may be, for example, a day of week decision node which compares the Event 1 call variable to value 1 in decision branch 1800c, which may be, for example, the values equal to Monday-Friday. If the call variable in Event 1 is equal to value 1, GetData node 1800d requires the

process described with respect to the CPR Editor screen 171 in Fig. 9.

The Custom Node Editor 791 also includes parameters window 797 which displays a list of parameters associated with the custom node being generated. These parameters define the relationship of the input fields for the custom node and the values within the graph. A parameter is a variable that will be filled in by the user of the custom node when it is inserted into a graph.

A Parameter Editor 1900, as shown for example in Fig. 19A, is used to create and modify parameters for a custom node. The Parameter Editor 1900 is displayed by "mouse clicking" on a preselected portion of the Parameters window 797. Parameter Editor 1900 prompts the user to complete a "parameter name" field 1900a, a "data type" field 1900b, an "allow" field 1900c, and an "interface" field 1900d. The parameter name is used when referring to this parameter as part of the value of a node. The "allow" field specifies permissible values for the parameter. For example, in Fig. 19A, the "allow" field 1900c permits only constants and call variables for the "Pin" parameter.

Using "Interface" field 1900d, the user can specify the type of interface to be displayed to a user of the customized node. Preferable interfaces include text fields, buttons, or selection lists. If a user designates the interface to be either buttons or selection lists, a Selection List Editor, as shown for example in Fig. 19B, is displayed.

The Selection List Editor 1902 allows the user to enter a list of labels which will be displayed when a custom node having the parameter being defined is used, as well as values associated with the labels.

The Selection List Editor 1902 includes a "Labels Defined In" field 1902a, a "Name" field 1902b, a "Label/Value" field 1902c, and a "Manipulators" field 1902d. Labels for a parameter may be defined in the Label/Value field 1902c or in another parameter. This allows a user to tie together the values of the parameters. Fields 1902a and

example in Fig. 21. As shown, the Layout Editor 2100 includes the same fields 2000a, 2000b, and 2000c, as displayed in the Preview Editor 2000. However, in the Layout Editor 2100, these fields can be manipulated by selecting a field (using select buttons 2100a) and clicking on one of the manipulator buttons 2100b.

The Set Category option 793 is used to establish a node category type for the custom node being created when a user selects the Set Category option 793, the system displays a Custom Node Category Editor 804, as shown for example in Fig. 22. Using the Custom Node Category Editor 804, a custom node may be assigned to any of the node types represented by the node type buttons 192 (Fig. 9).

When the custom node is fully defined and categorized, the user enables the node by selecting an "Enable" suboption (not shown) from the "Operations" menu (not shown) on the System screen 170 (Fig. 7). Preferably, the underlying graph is validated prior to being enabled. Once a User-defined node has been enabled, it will appear in the nodes list 191 of the CPR Editor screen 171 and the nodes lists 126a and 128a of the GSS Editor screen 120.

When a CPR containing a custom node is trace tested, the custom node will be displayed as a single node. In other words, the underlying graph is not displayed. However, individual nodes within the underlying graph of the custom node are tested in the same manner as other nodes in the graph. Each node of the underlying graph of a custom node is also considered during validation. Thus, errors and warnings generated by a testing or validation process can be specified to a particular node within the underlying graph of the custom node.

displayed when a user selects an Intable node from a nodes list.

Intable Node Editor 2200 includes Name field 2200a corresponding to this node type. The table search criteria is inserted in search fields 2200b-e. Table Name field 2200b specifies the table to be searched. Column field 2200c specifies the column or columns of the table to be searched. Value field 2200d specifies a value to be searched for in the specified column. Finally, Expression field 2200e permits a user to specify comparison criteria for the value specified in field 2200d and the values in the table. In a preferred embodiment, the comparison criteria in the Expression field 2200e includes "=," "=," ">," "<," "≥," and "≤."

In a preferred implementation of the present invention, a method by which the system executing an Intable node searches a designated table and outputs a response is illustrated in the flowchart of Fig. 26. Initially, when executing a table node the system reads the Table name designated by the Intable node (step 1230) and determines whether such a table exists (step 1231). If not, an error is indicated (step 1235). If the table is found, however, the system reads the Column names to be searched (step 1232) and determines whether the Columns exist in the Table (step 1233). If not, an error is indicated (step 1235). Once the Table and Columns are found, the system reads the value(s) to be searched (step 1236), and searches the Table Columns using the expression contained in the Intable node to compare the specified values to values in the Table (step 1237). If the value(s) are found in the Table, the call is processed one way; if the value(s) are not found in the Table, the call is processed another way, as designated by the branches in the graph.

user specifies the column name or names of a table from which to retrieve a value and the corresponding call variable name or names to which the retrieved value(s) should be assigned. Additional column names and call variable names can be added or deleted using manipulator buttons 2300f.

Upon execution of a graph having a Table node, the call variables designated by the TABLE node will have either values obtained from the table designated or null values.

A preferred method by which the system executing a graph having a TABLE node searches a designated table and outputs a response is illustrated in the flowchart of Fig. 28. Initially, the system sequentially reads the call variables designated in the Table node (step 1250), the table name designated by the Table node (step 1252), and the Column names designated in the Table node (step 1254). After reading each of these designations, the system respectively determines whether each exists (steps 1251, 1253, and 1255). If one does not exist, an error is indicated (step 1256). Once the call variables, table, and column names have been read, the system reads the search values (step 1256) and searches the Table using the comparison expressions contained in the Table node (step 1257). If values are found in the columns which meet the requirements of the search values, the values are output (step 1259). If no such values are found, "null" values are output (step 1260).

G. Templates

Many customers may request the same telecommunication service for mass markets. For example, many customers may wish to designate a long distance carrier during certain times of the day (i.e., business hours). Each customer's graph would therefore be identical except for call variables and nodes and branches defining the carriers and nodes defining the time of day that specified carriers will service the call. All other nodes in the graph and the structure of the graph would be "generic" to the service.

Template Editor screen 910 includes a Template Record Information window 911, a Call Variables window 913, a Graphs In Template window 912, and a Form Operations window 914. The Template Record Information window 911 includes "Name," "Creator," "Modified," and "Effective" fields 911a-d, similar to these same fields for the CPR, GSS, and Custom nodes screens (see Figs. 9, 10, and 18). The Graphs In Template window 912 and Call Variables window 913 of the Template Editor screen 910 operate in the same way as the Graphs in CPR window 178 and the Call Variables window 190, respectively, of the CPR Editor screen 171 (Fig. 9). Form Operations window buttons 914a and 914b are described below.

The graph 925 from which the template is being created is displayed in Graph window 920. The exemplary graph of Fig. 29A provides for a predetermined carrier for all calls made to a particular telephone number and routes the calls to one of two telephone numbers depending on whether the calls are made on a weekday or weekend. A user from which the template is being created can select which of the nodes of the graph he or she wishes to make customizable by clicking a mouse or similar device on the node.

Each expression in the selected node can be designated as customizable. For example, assume that the template creator selects the "Carrier" node 925a to be customizable. In response to this selection, the system displays a Template Carrier Node Editor 930. Template Node Editors in general differ from CPR Node Editors because Template Node Editors include customizable selection buttons 935, which allow a user to designate which node expressions will be customizable. For example, in Fig. 29B, the carrier type is not customizable, but is fixed as primary. However, the carrier value is customizable. Text fields 936a and 936b are provided to specify a prompt which will be displayed to a user to collect the customizable information for the node.

In like manner, to make a branch a call variable customizable, in response to a selection of the branch or call variable by the user, the system prompts the user to

Selection of the "Layout" option 914a causes the system to display a Layout Editor 916, as shown for example in Fig. 29D. The Layout Editor 916 includes the same fields 915a-d as shown in the Preview Editor 915, and shows the layout of information that will be presented to a user creating a CPR based on a particular template. A set of manipulator buttons 916a is provided to allow the user to change the order of the fields. Preferably, only the order of the entry fields is changed in the Layout Editor 916.

After the user makes "customizable" all the nodes required to transform the CPR graph 925 into an appropriate template, the user enables the template by selecting an Enable suboption (not shown) from the main menu bar "Operations" Menu (not shown). The enabled template is then available for making template-based CPRs and can be stored in the database 203.

A user creates a template-based CPR by selecting the "Find Template" option 178 under the Record menu of the main menu bar 172. Selection of the "Find Template" option causes the system to display a Find Editor 950, as shown for example in Fig. 30, which displays in list window 950a a list of templates stored in database 203. For each template stored in database 203, the system displays the name, status, and creator of the status, as well as dates the template was enabled and modified. Find Editor 950 also includes search fields 950b, which allow a user to designate search criteria to search the template list. Menu buttons 950c permit a user to edit, browse, delete, customize, or cancel a selected template.

A user selects a template by selecting the template name (e.g., mouse click) in the template list 950a and selecting the customize button. In response to these selections, the system displays a New Record Information Dialog Box requesting the user to input a name of the template-based CPR. The user then has the option of viewing the template-based CPR in a graph representation (which looks like the graph 925 shown in Fig. 29A) or in a form representation

WHAT IS CLAIMED:

1. A method of creating, in response to inputs from an operator of a record creation system in a telecommunication network, a general service specification for a call processing record containing logically related nodes and branches, the method comprising the steps, executed by a processor in the record creation system, of:

prompting the operator to identify at least one optional node which may appear in a call processing record associated with the general service specification;

receiving from the operator an identification of at least one optional node which may appear in the call processing record associated with the general service specification;

prompting the operator to identify at least one required node which must appear in the call processing record associated with the general service specification;

receiving from the operator an identification of at least one required node, which must appear in call processing records associated with the general service specification; and

enabling said at least one optional node and said at least one required node as a general service specification.

2. A method according to claim 1, further comprising the steps of:

prompting the operator to identify at least one restricted node which cannot appear in the call processing record associated with the general service specification;

receiving from the operator an identification of at least one restricted node which cannot appear in the call processing record associated with the general service specification; and

enabling at least one restricted node as part of the general service specification.

3. A method of creating, in response to inputs from an operator of a record creation system, a call processing

receiving from the operator an identification of a selected node in the call processing record to be made customizable, a customizable node being a node for which subsequent template users can specify predetermined expressions;

displaying to the operator all expressions of the selected node;

prompting the operator to identify which of the selected node expressions will be customizable;

receiving from the operator an identification of an expression of the selected node which will be customizable; and

enabling the selected call processing record and the designation of customizable node expressions for the selected node as a service template.

6. The method according to claim 5, further comprising the step of displaying the service template as a graph representation or a form representation.

7. A method of creating, in response to inputs from an operator of a record creation system, a template for the creation of call processing services, each call processing service being represented by a call processing record containing logically related call processing nodes, branches, and call variables, the method comprising the steps, executed by a processor, of:

displaying to the operator a selected call processing record;

receiving from the operator an identification of a selected branch in the call processing record to be made customizable, a customizable branch being a branch for which subsequent template users can specify predetermined expressions;

displaying to the operator all expressions of the selected branch;

prompting the operator to identify which of the selected branch expressions will be customizable;

variable expression, the method comprising the steps, executed by a processor, of:

retrieving the service template from the database;

displaying a representation of the retrieved service template;

prompting the operator to provide information to specify at least one variable expression of the at least one customizable node;

defining the variable expression of the at least one customizable node with the information provided by the operator; and

enabling the displayed representation of the retrieved service template and defined expression as a call processing record.

10. A method of providing a requested service to one or more customers of a telecommunication network, the method comprising the steps, executed by a data processor of the telecommunication network, of:

creating one or more call processing records each including a plurality of call processing procedures for execution by a call processing environment of the telecommunication network;

creating a table of data associated with each of said one or more call processing records;

storing said one or more call processing records and said table of data;

executing one of said processing records in the call processing environment; and

retrieving data from said table of data during the execution of said one of said call processing records.

11. In a telecommunication service creation environment in a telecommunication network providing for call processing records and value tables, each of the value tables comprising one or more columns and one or more rows of values, a method of creating a call processing procedure to determine whether a particular value exists in a particular value table comprising the steps, executed by a data processor, of:

comparing values in the one or more columns to the specified search value in accordance with the specified comparison criteria;

generating a first output if the comparison criteria is met during the comparing step; and

generating a second output if the comparison criteria is not met during the comparing step.

13. In a telecommunication service creation environment providing for call processing records and value tables, the value tables comprising one or more columns and one or more rows of values, a method of creating a call processing procedure to retrieve a value from the value table for call processing, the method comprising the steps, executed by a data processor, of:

prompting an operator to name a value table to be searched;

receiving from the operator a name of the value table to be searched;

prompting an operator to identify one or more columns in the value table to be searched;

receiving from the operator on identification of one or more values in the value table to be searched;

prompting an operator to specify a value to be searched for in the one or more columns to be searched;

receiving from the operator a value to be searched for in the one or more columns to be searched;

prompting an operator to specify comparison criteria for the value specified and values in the column to be searched;

receiving from the operator a comparison criteria for the value specified and values in the column to be searched;

prompting an operator to specify one or more call variable names to which one or more retrieved values should be assigned;

receiving a sampling type for said sample node, said sampling type defining whether the activity should be determined based on attempts or completions;

receiving a sample name for said sample node, said sample name defining a name for data collected; and

receiving a list of call variables to be collected.

16. A method of designing a procedure to direct a telecommunication network to provide requested services to an individual customer of the network, the method comprising the steps, executed by a data processor, of:

presenting the customer with a plurality of types of nodes, the nodes indicating the determinations and actions allowable for the procedure;

receiving from the customer indications of desired nodes;

receiving from the customer indications of desired relationships between the desired nodes;

receiving from the customer values for parameters to be used with the desired nodes; and

construction of a graphical representation of the desired nodes reflecting the customer values and the indicated relationships among the nodes, wherein one of said nodes comprises a measurement node for counting a predetermined call processing event.

17. A method according to claim 16, wherein said step of receiving from the customer values for parameters to be used with the desired nodes includes the steps of:

receiving a call variable naming a measurement vector;

receiving a component name identifying a component within the measurement vector which will be incremented or decremented; and

receiving information specifying when the measurement vector should be incremented or decremented.

18. A method of providing a call processing measurement node to count call processing events, the method comprising the steps, executed by a processor, of:

processing procedures, said parameters defining call variables for which values can be provided at a later time.

22. The method according to claim 21, further comprising the step of receiving from the operator a parameter name, data type, allowed inputs, and interface type.

23. A call processing record for execution in a telephone service execution environment, comprising:

at least one call processing logic section including call processing procedures executable by a processor in said telephone service execution environment;

at least one first data section, each of said at least one first data sections being associated with one of said at least one call processing logic sections and storing data executable only by said call processing procedures included in the associated one of said at least one call processing sections; and

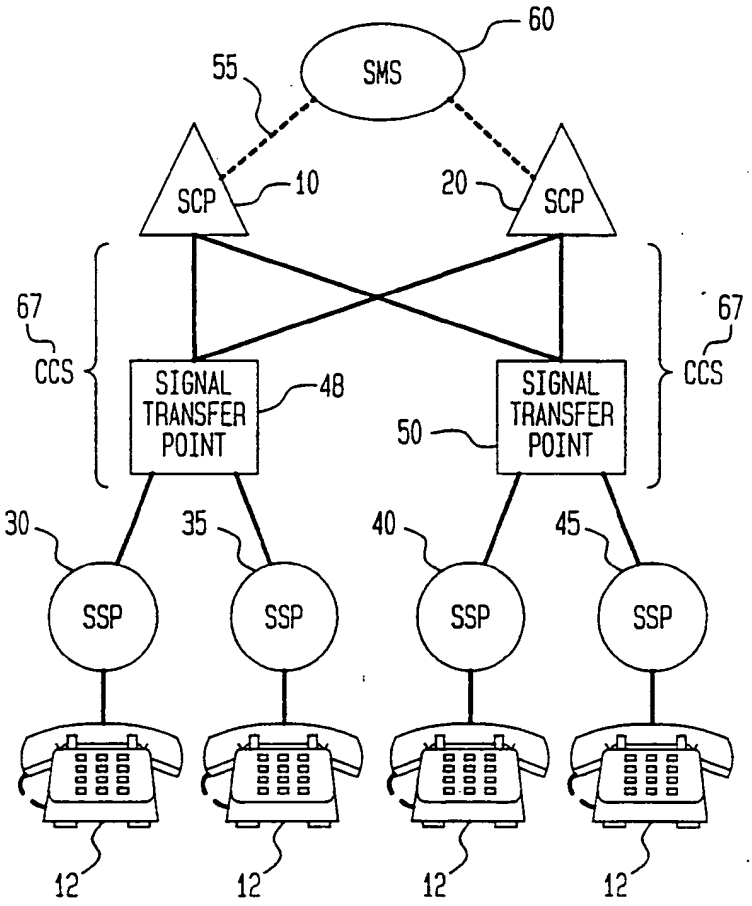
at least one entry point, each of said at least one entry points being associated with one of said at least one call processing logic sections and an associated one of said at least one first data sections, said at least one entry point identifying the associated one of said at least one call processing sections.

24. A call processing record according to claim 23, further comprising a second data section including data executable by call processing procedures in each of said at least one call processing logic sections.

25. A call processing record according to claim 23, further comprising a record header identifying said call processing record and including a telephone number for the corresponding telephone service subscriber.

26. A call processing record according to claim 23, wherein one of said at least one entry points comprises a trigger identifying a telephone call either originating from a called telephone number or being made to a called telephone number.

FIG. 1
(PRIOR ART)



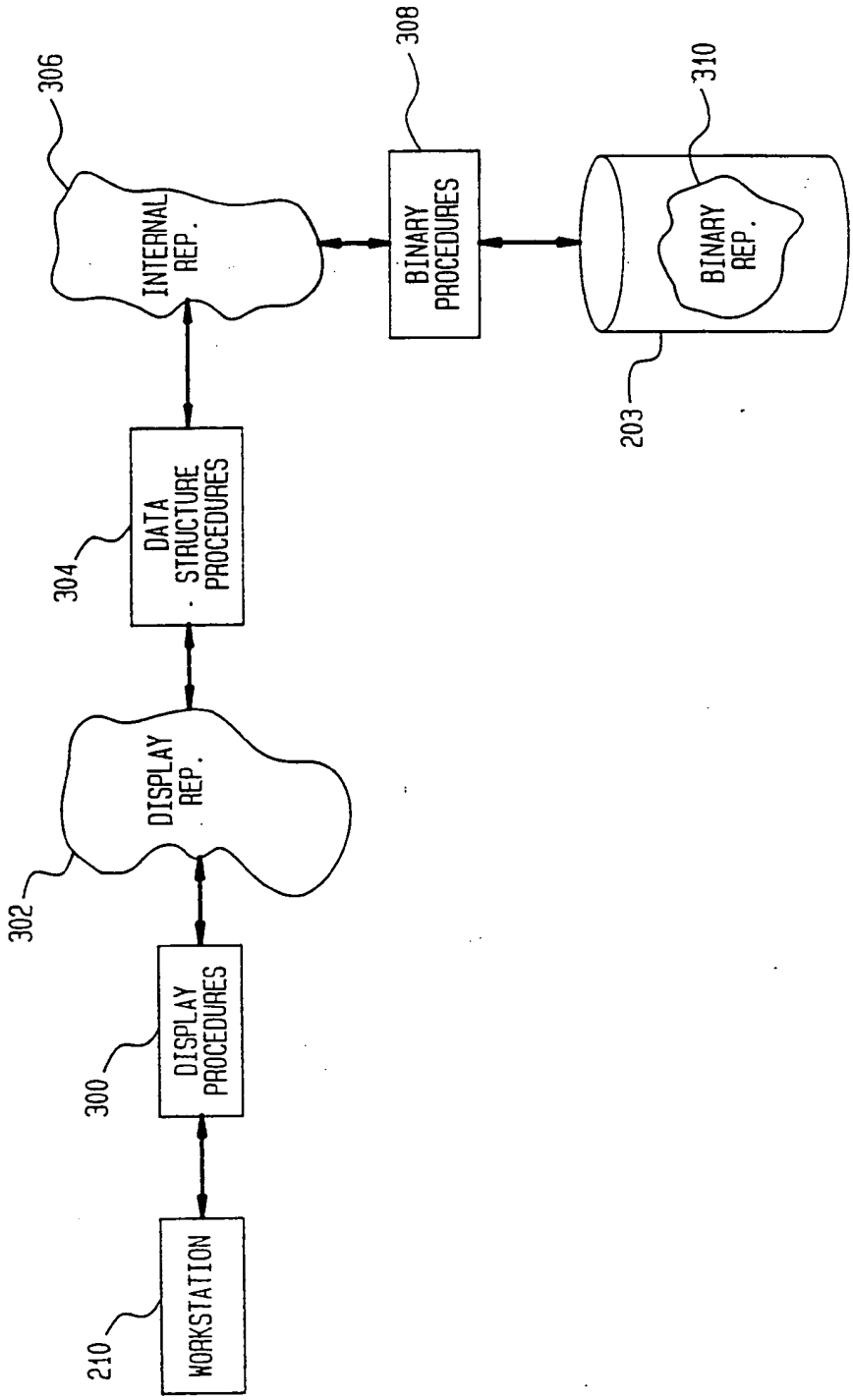

306
INTERNAL REP.
308
BINARY PROCEDURES
310
BINARY REP.
203
304
DATA STRUCTURE PROCEDURES
302
DISPLAY REP.
300
DISPLAY PROCEDURES
210
WORKSTATION

FIG. 3

FIG. 4B

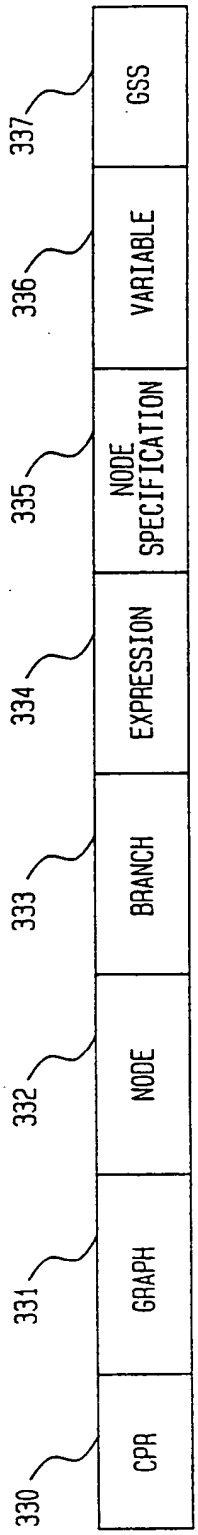

330
CPR
331
GRAPH
332
NODE
333
BRANCH
334
EXPRESSION
335
NODE SPECIFICATION
336
VARIABLE
337
GSS

FIG. 4C

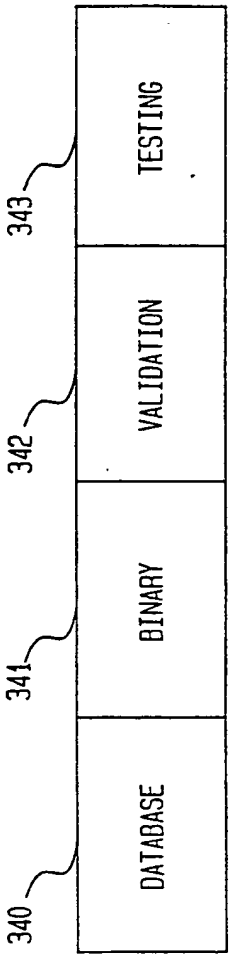

340
DATABASE
341
BINARY
342
VALIDATION
343
TESTING

FIG. 6A

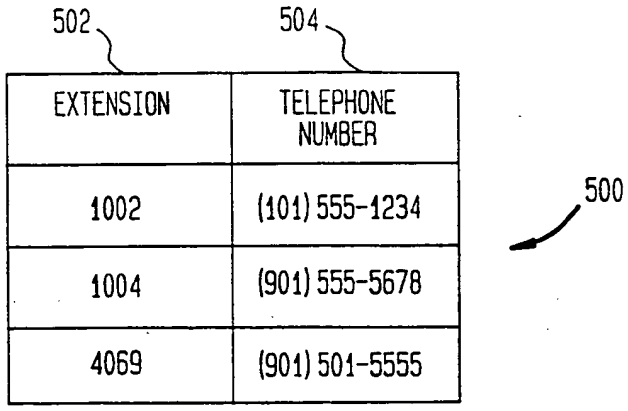


FIG. 6B

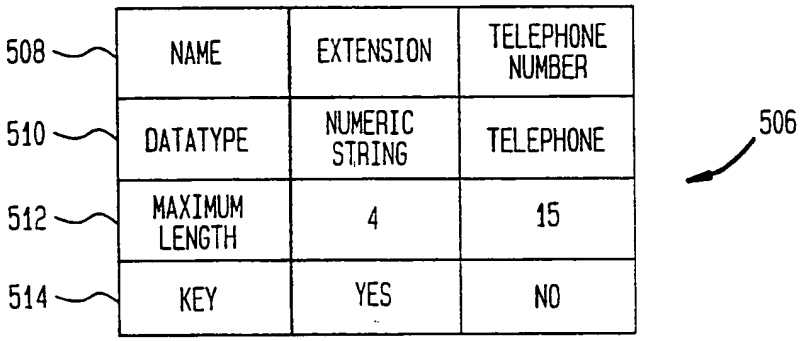


FIG. 6C

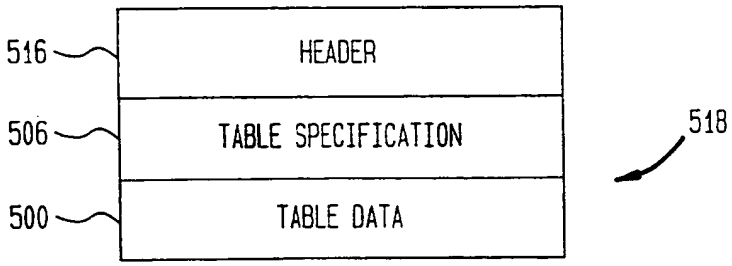


FIG. 8

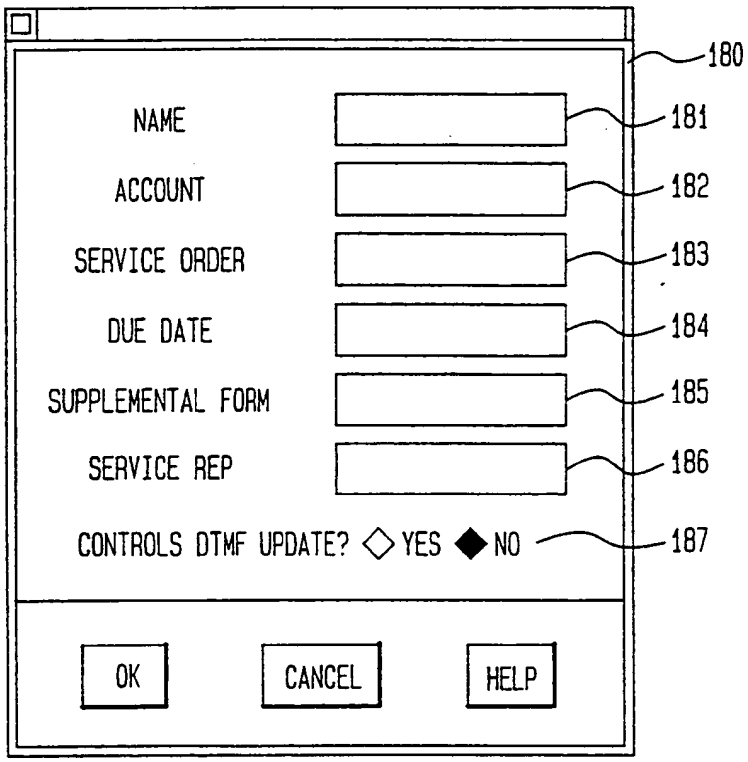

180
NAME
181
ACCOUNT
182
SERVICE ORDER
183
DUE DATE
184
SUPPLEMENTAL FORM
185
SERVICE REP
186
CONTROLS DTMF UPDATE?
YES
NO
187
OK
CANCEL
HELP

FIG. 10

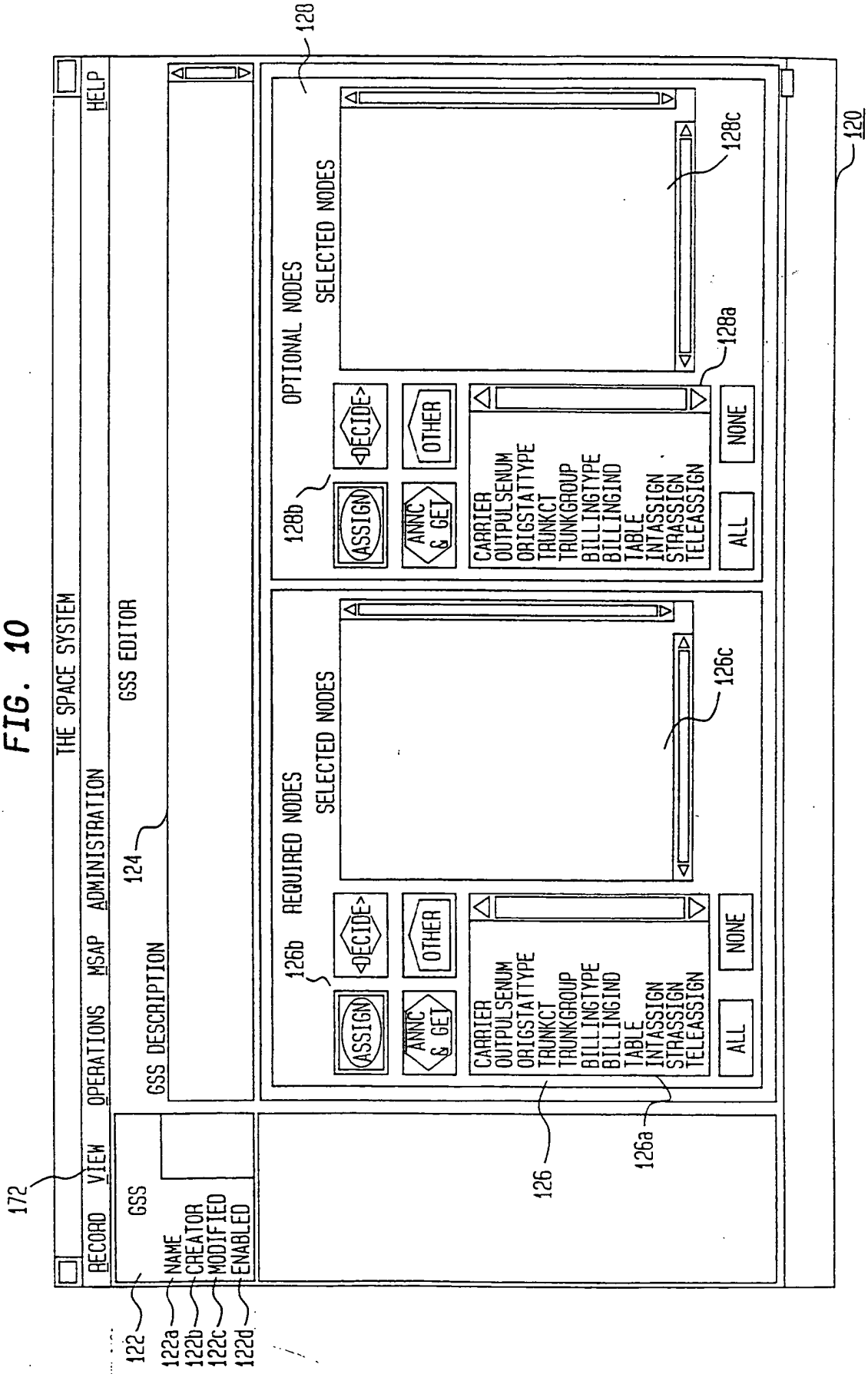

THE SPACE SYSTEM
RECORD VIEW OPERATIONS MSAP ADMINISTRATION HELP
GSS EDITOR
GSS DESCRIPTION
GSS
NAME
CREATOR
MODIFIED
ENABLED
REQUIRED NODES
SELECTED NODES
OPTIONAL NODES
SELECTED NODES
ASSIGN
ANNC & GET
<DECIDE>
OTHER
CARRIER
OUTPULSENUM
ORIGSTATTYPE
TRUNKCT
TRUNKGROUP
BILLINGTYPE
BILLINGIND
TABLE
INTASSIGN
STRASSIGN
TELEASSIGN
ALL
NONE
120
122
122a
122b
122c
122d
124
126
126a
126b
126c
128
128a
128b
128c
172

FIG. 12

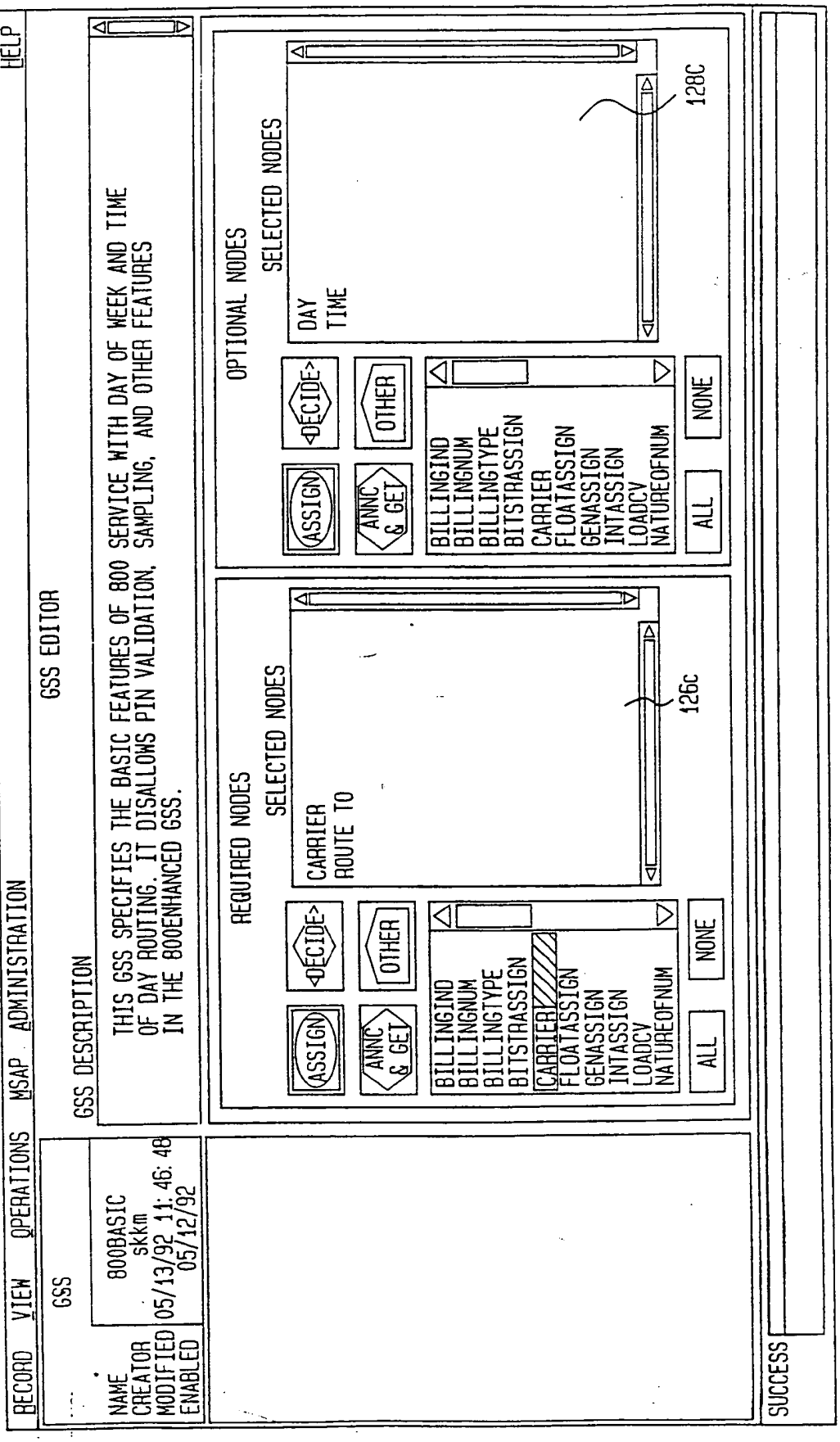

120
RECORD VIEW OPERATIONS MSAP ADMINISTRATION HELP
GSS EDITOR
GSS
NAME
CREATOR
MODIFIED
ENABLED
800BASIC
skkm
05/13/92 11:46:48
05/12/92
GSS DESCRIPTION
THIS GSS SPECIFIES THE BASIC FEATURES OF 800 SERVICE WITH DAY OF WEEK AND TIME OF DAY ROUTING. IT DISALLOWS PIN VALIDATION, SAMPLING, AND OTHER FEATURES IN THE 800ENHANCED GSS.
REQUIRED NODES
SELECTED NODES
CARRIER
ROUTE TO
ASSIGN
<DECIDE>
ANNC & GET
OTHER
BILLINGIND
BILLINGNUM
BILLINGTYPE
BITSTRASSIGN
CARRIER
FLOATASSIGN
GENASSIGN
INTASSIGN
LOADCV
NATUREOFNUM
ALL
NONE
126c
OPTIONAL NODES
SELECTED NODES
DAY
TIME
128C
SUCCESS

FIG. 14

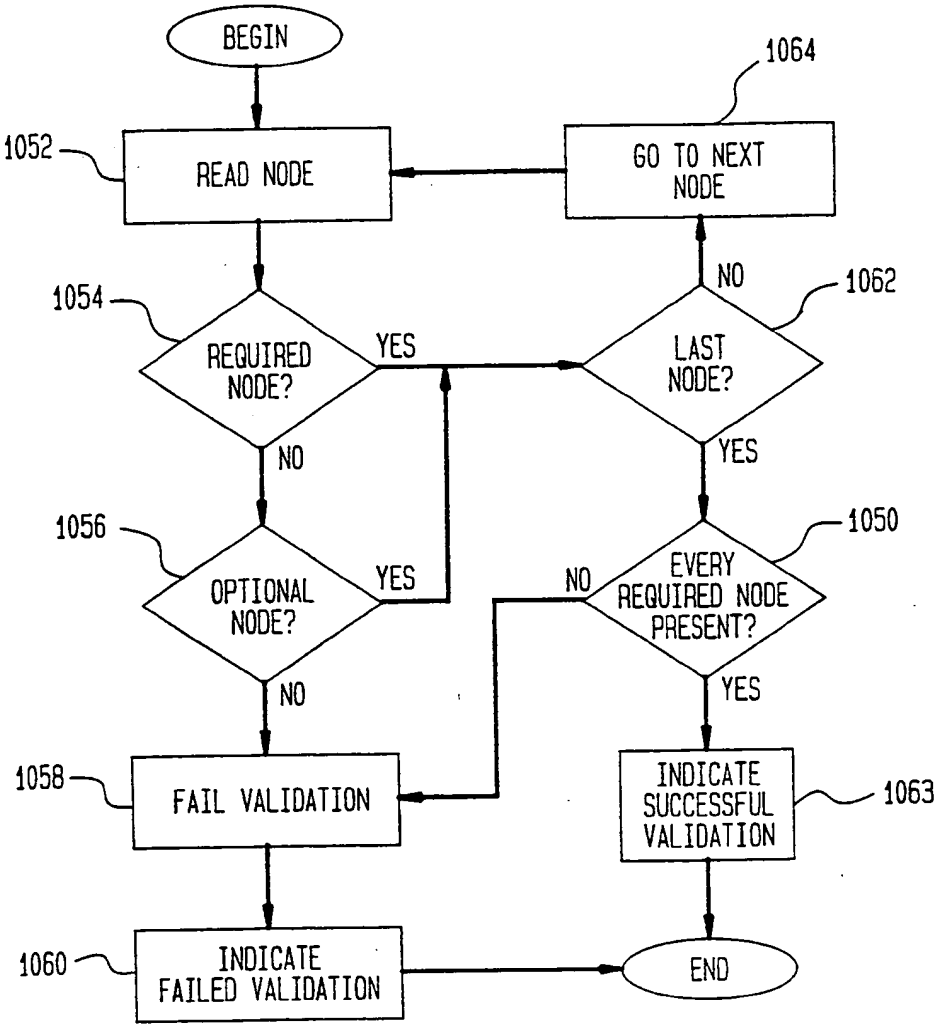

BEGIN
1064
1052
READ NODE
GO TO NEXT NODE
NO
1062
1054
REQUIRED NODE?
YES
LAST NODE?
NO
YES
1056
1050
OPTIONAL NODE?
YES
NO
EVERY REQUIRED NODE PRESENT?
NO
YES
1058
FAIL VALIDATION
INDICATE SUCCESSFUL VALIDATION
1063
1060
INDICATE FAILED VALIDATION
END

FIG. 16

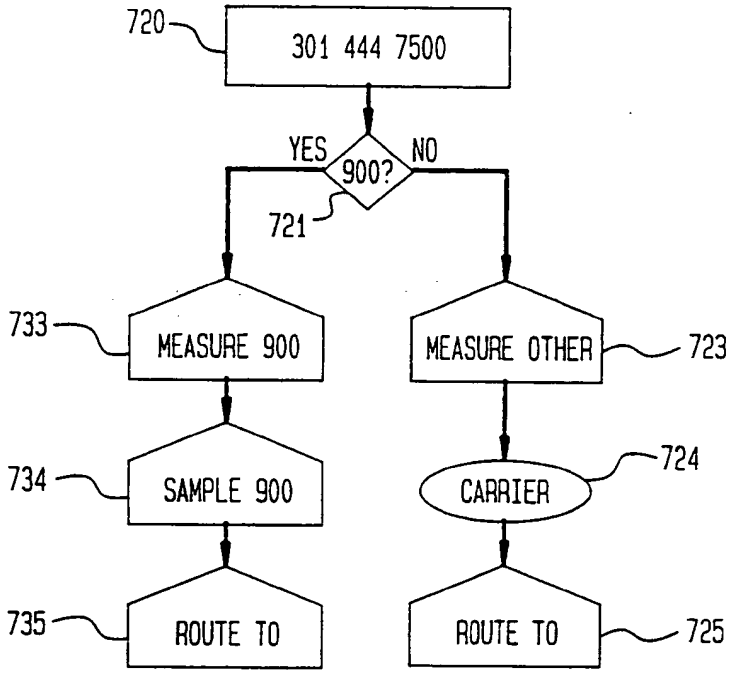

720
301 444 7500
YES
900?
NO
721
733
MEASURE 900
MEASURE OTHER
723
734
SAMPLE 900
CARRIER
724
735
ROUTE TO
ROUTE TO
725

FIG. 17

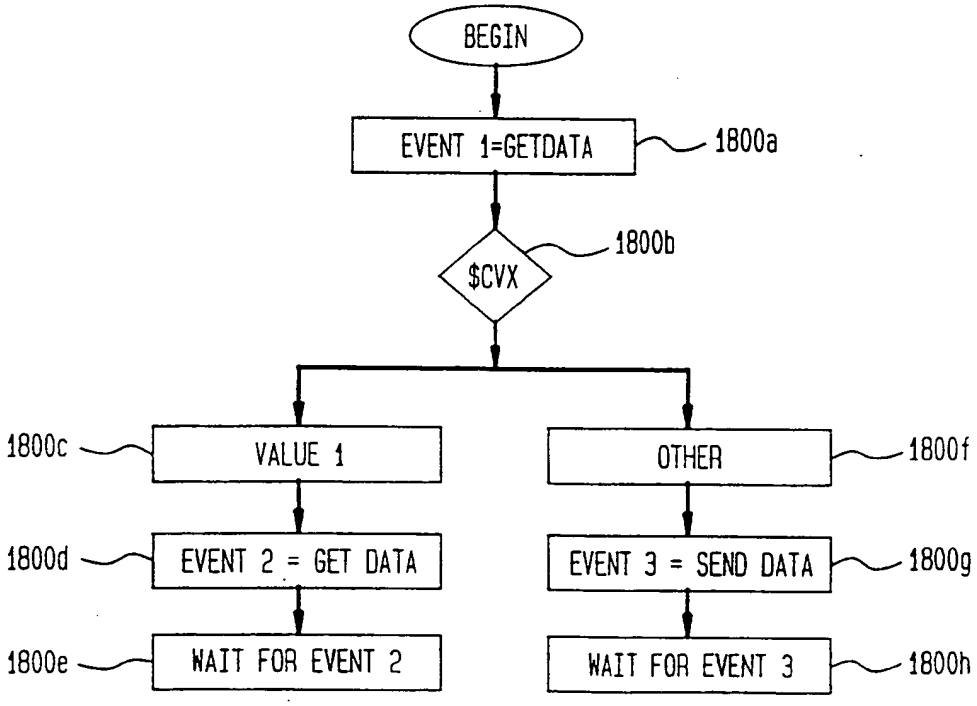

BEGIN
EVENT 1=GETDATA
1800a
$CVX
1800b
1800c
VALUE 1
OTHER
1800f
1800d
EVENT 2 = GET DATA
EVENT 3 = SEND DATA
1800g
1800e
WAIT FOR EVENT 2
WAIT FOR EVENT 3
1800h

FIG. 19A

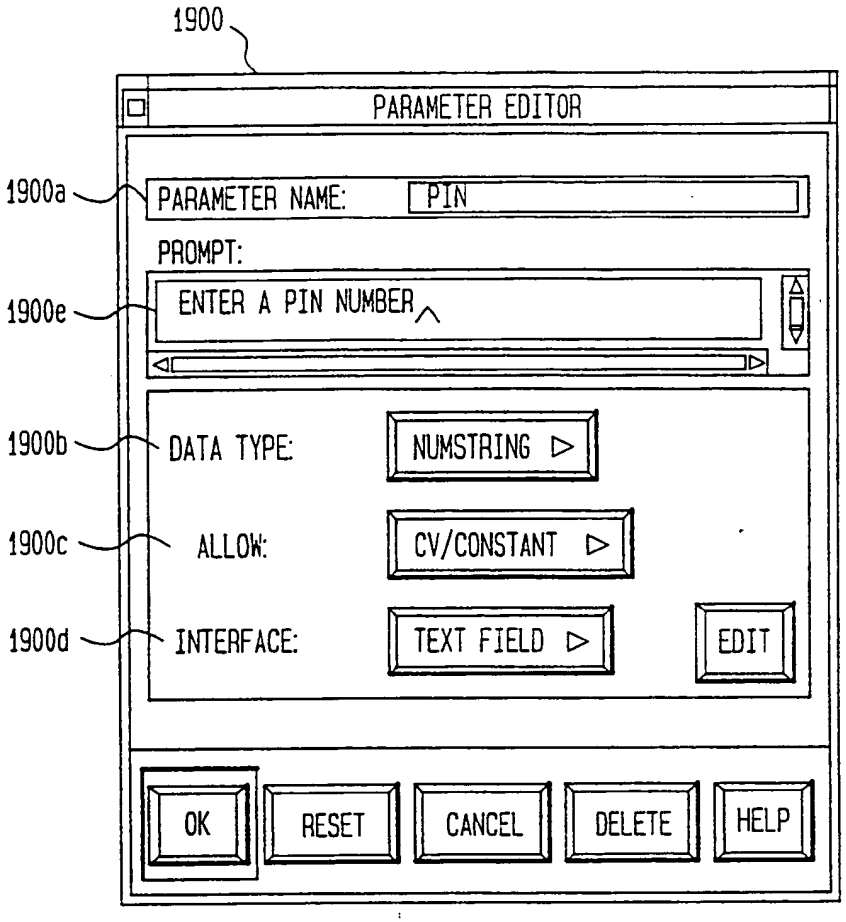


FIG. 19B

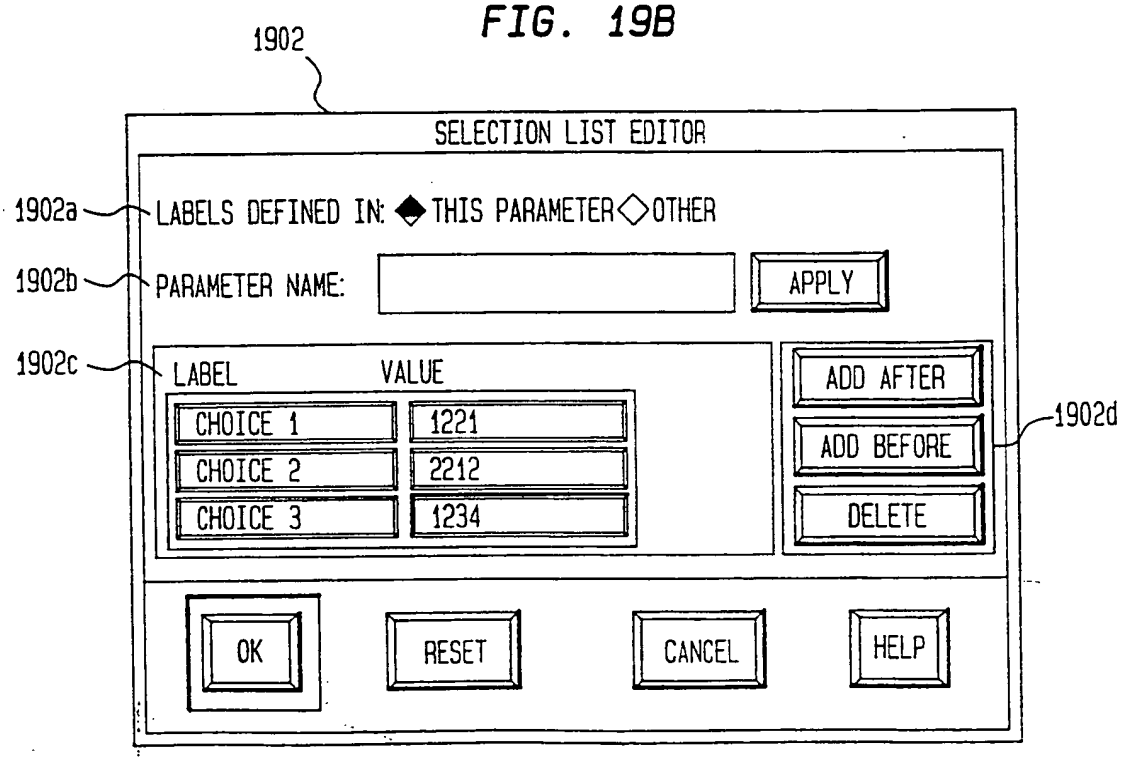


FIG. 21

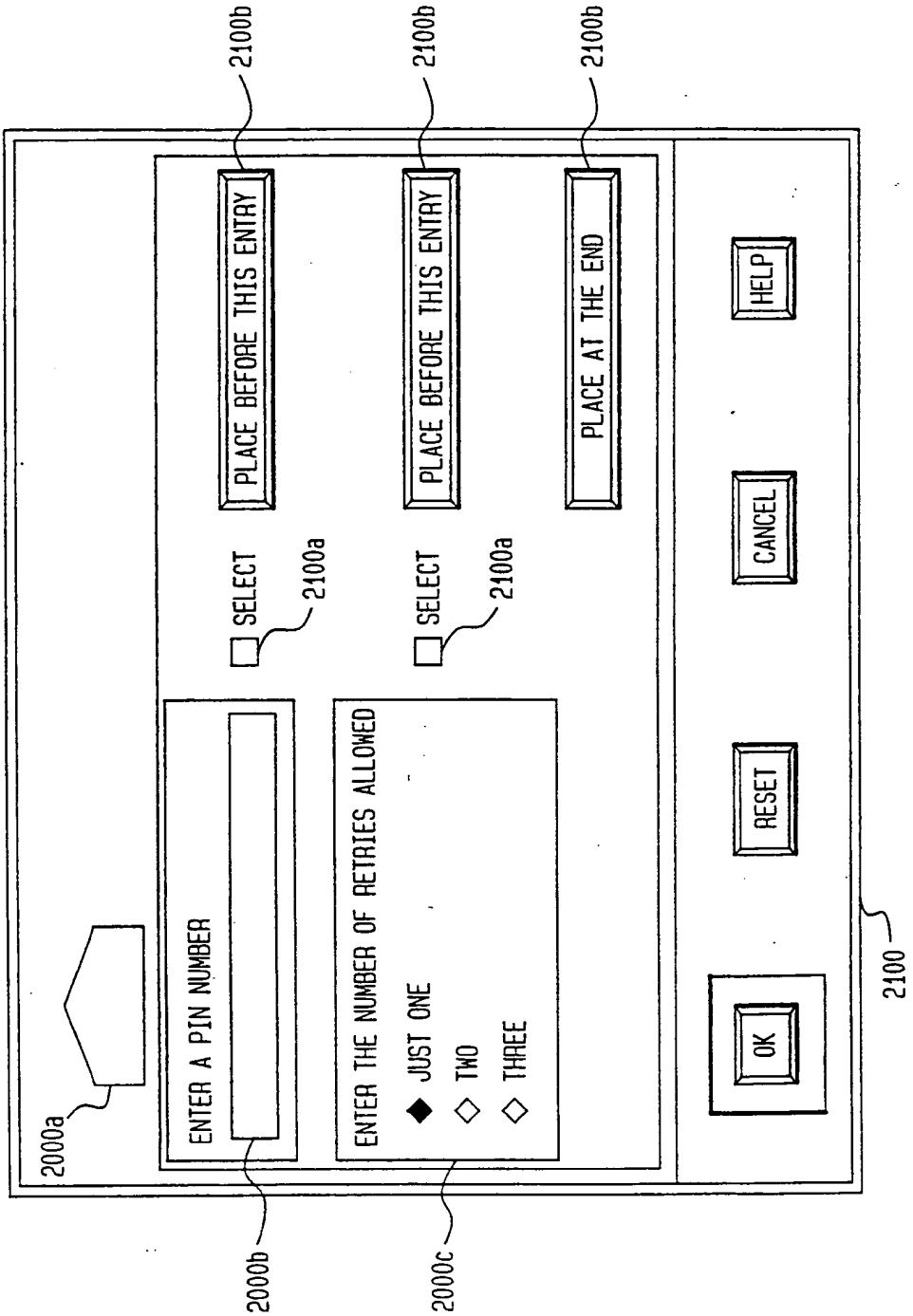

2100b
2100b
2100b
PLACE BEFORE THIS ENTRY
PLACE BEFORE THIS ENTRY
PLACE AT THE END
HELP
SELECT
2100a
SELECT
2100a
CANCEL
ENTER A PIN NUMBER
ENTER THE NUMBER OF RETRIES ALLOWED
JUST ONE
TWO
THREE
RESET
OK
2000a
2000b
2000c
2100

FIG. 25

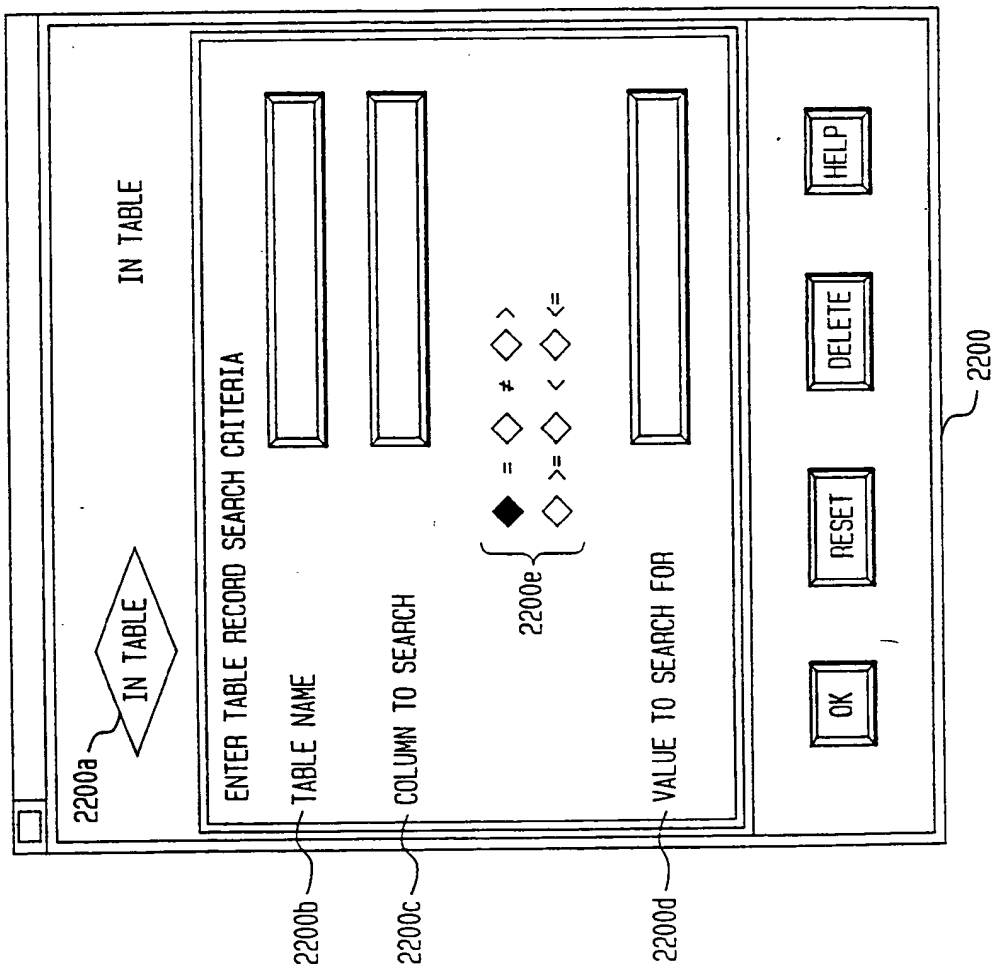

IN TABLE
2200a
IN TABLE
ENTER TABLE RECORD SEARCH CRITERIA
TABLE NAME
COLUMN TO SEARCH
= ≠ >
>= < <=
2200e
VALUE TO SEARCH FOR
HELP
DELETE
RESET
OK
2200
2200b
2200c
2200d

FIG. 27

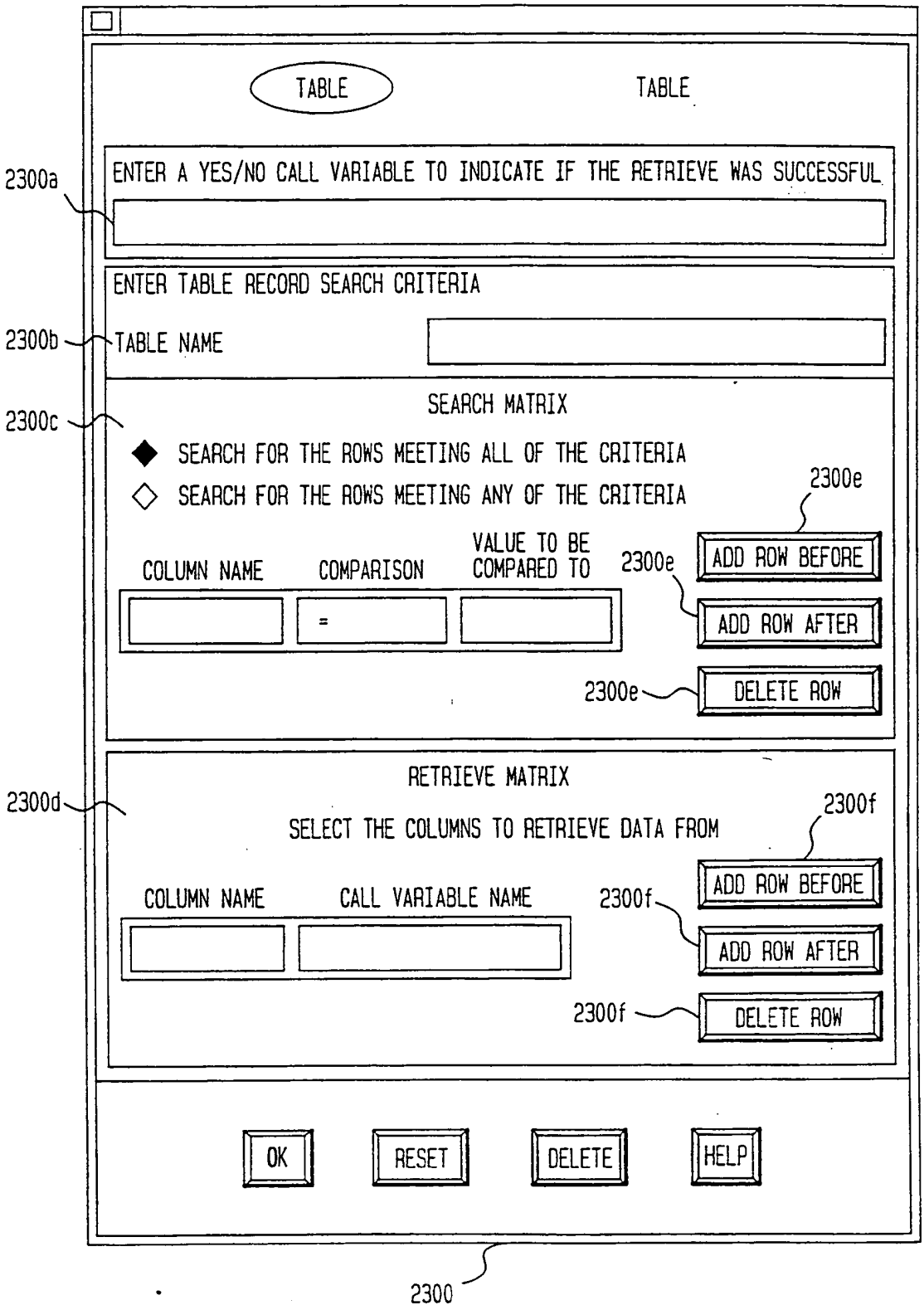


TABLE
TABLE
2300a
ENTER A YES/NO CALL VARIABLE TO INDICATE IF THE RETRIEVE WAS SUCCESSFUL
ENTER TABLE RECORD SEARCH CRITERIA
2300b
TABLE NAME
2300c
SEARCH MATRIX
SEARCH FOR THE ROWS MEETING ALL OF THE CRITERIA
SEARCH FOR THE ROWS MEETING ANY OF THE CRITERIA
2300e
COLUMN NAME
COMPARISON
VALUE TO BE COMPARED TO
=
2300e
ADD ROW BEFORE
ADD ROW AFTER
2300e
DELETE ROW
2300d
RETRIEVE MATRIX
SELECT THE COLUMNS TO RETRIEVE DATA FROM
2300f
COLUMN NAME
CALL VARIABLE NAME
2300f
ADD ROW BEFORE
ADD ROW AFTER
2300f
DELETE ROW
OK
RESET
DELETE
HELP
2300

FIG. 29A

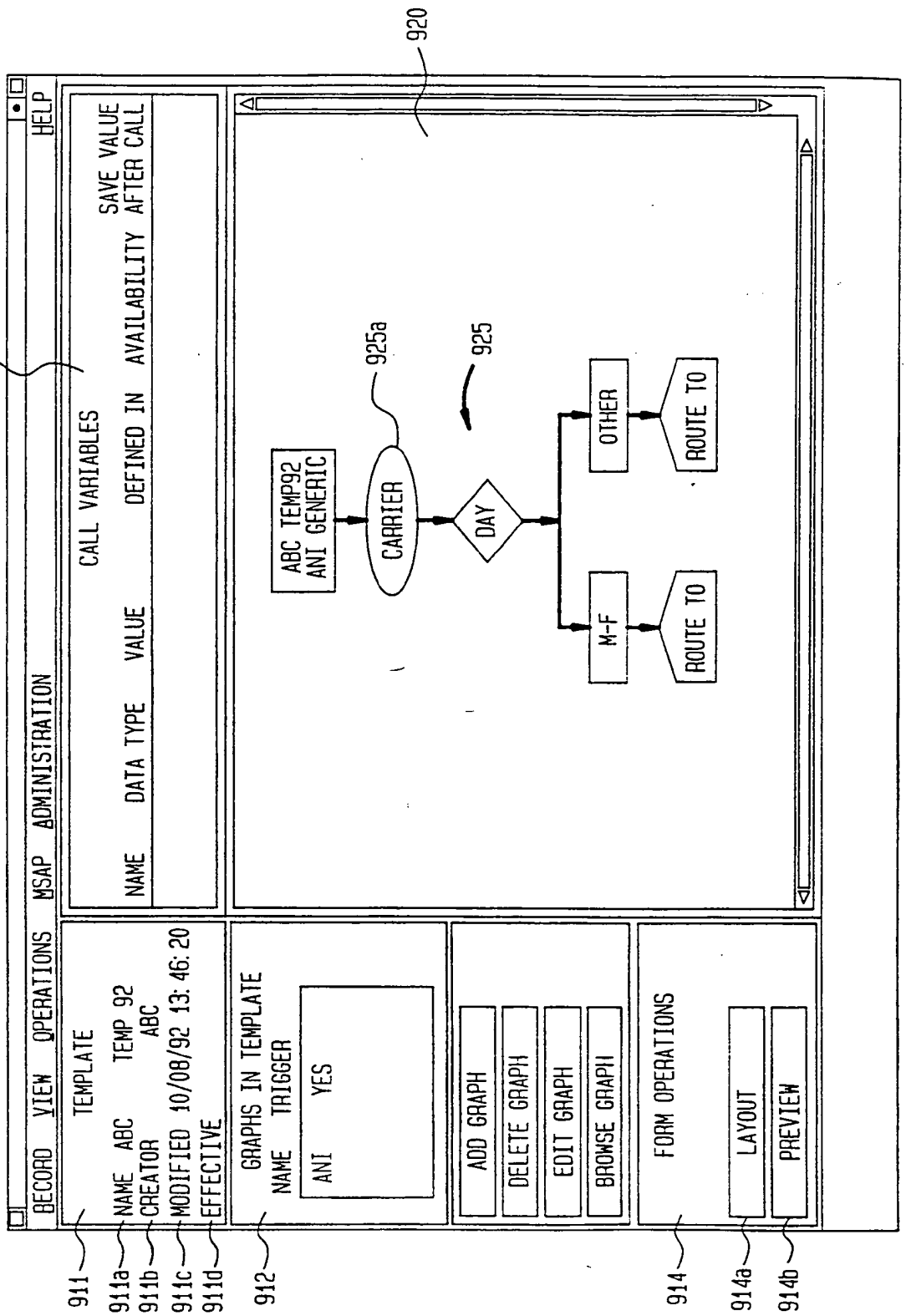
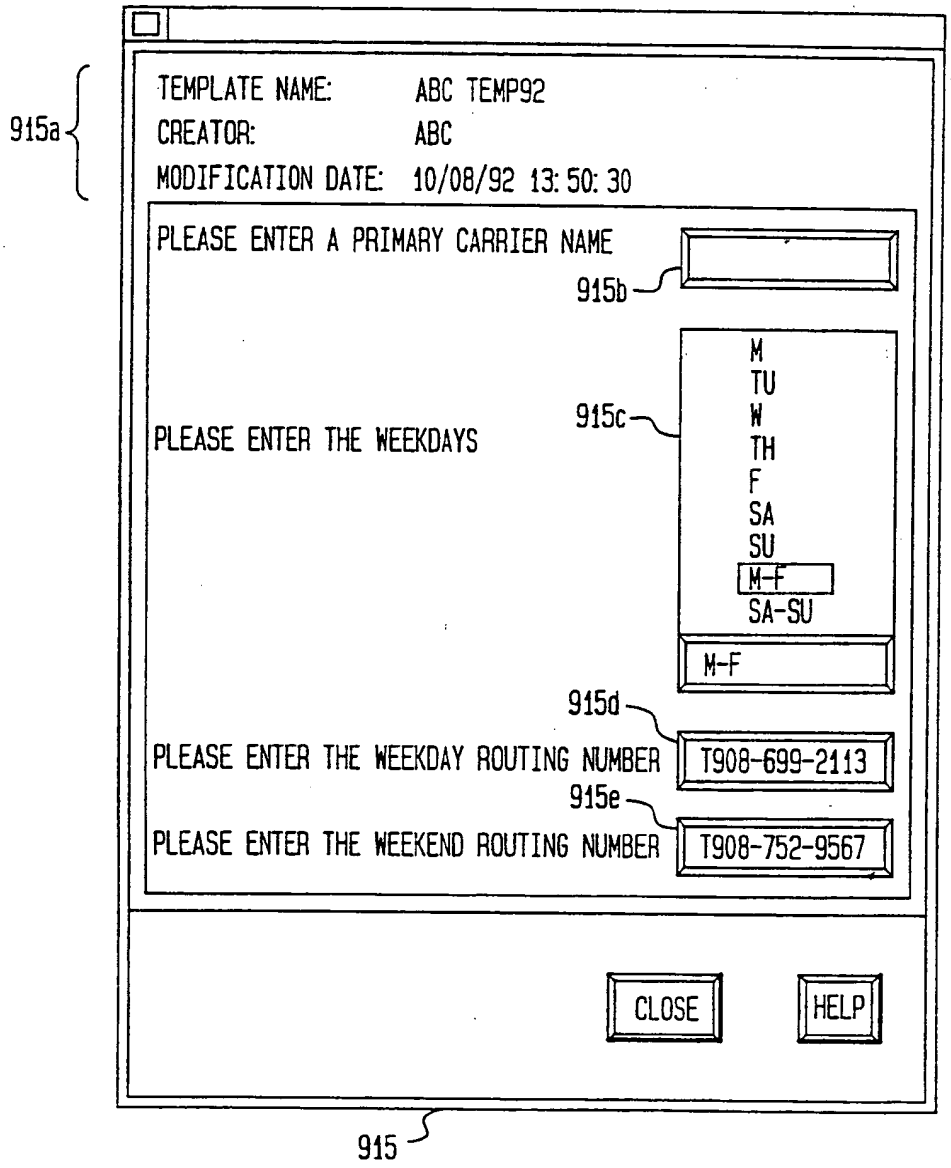

913
920
925a
925
RECORD VIEW OPERATIONS MSAP ADMINISTRATION HELP
CALL VARIABLES
NAME DATA TYPE VALUE DEFINED IN AVAILABILITY SAVE VALUE AFTER CALL
TEMPLATE
NAME ABC TEMP 92
CREATOR ABC
MODIFIED 10/08/92 13:46:20
EFFECTIVE
GRAPHS IN TEMPLATE
NAME TRIGGER
ANI YES
ADD GRAPH
DELETE GRAPH
EDIT GRAPH
BROWSE GRAPH
FORM OPERATIONS
LAYOUT
PREVIEW
ABC TEMP92
ANI GENERIC
CARRIER
DAY
M-F
ROUTE TO
OTHER
ROUTE TO
911
911a
911b
911c
911d
912
914
914a
914b

FIG. 29C


915a
TEMPLATE NAME: ABC TEMP92
CREATOR: ABC
MODIFICATION DATE: 10/08/92 13:50:30
PLEASE ENTER A PRIMARY CARRIER NAME
915b
PLEASE ENTER THE WEEKDAYS
915c
M
TU
W
TH
F
SA
SU
M-F
SA-SU
M-F
915d
PLEASE ENTER THE WEEKDAY ROUTING NUMBER
T908-699-2113
915e
PLEASE ENTER THE WEEKEND ROUTING NUMBER
T908-752-9567
CLOSE
HELP
915

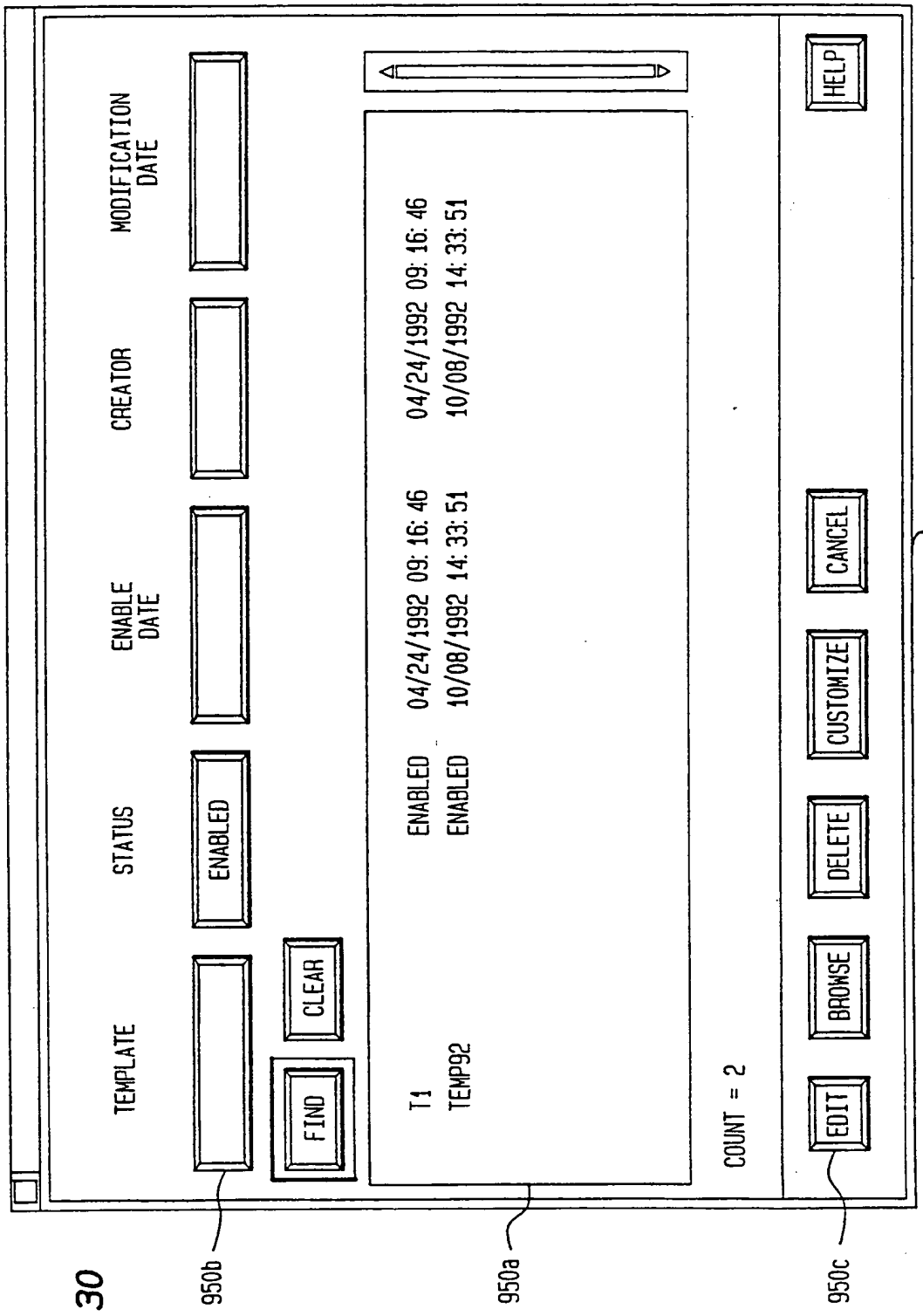

TEMPLATE
STATUS
ENABLE DATE
CREATOR
MODIFICATION DATE
ENABLED
FIND
CLEAR
T1
TEMP92
ENABLED 04/24/1992 09: 16: 46 04/24/1992 09: 16: 46
ENABLED 10/08/1992 14: 33: 51 10/08/1992 14: 33: 51
COUNT = 2
EDIT
BROWSE
DELETE
CUSTOMIZE
CANCEL
HELP
950
950a
950b
950c

FIG. 30

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US93/07835

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	IEEE Communications Magazine, January, 1991, Masanobu Fujioka et al., "Universal Service Creation and Provision Environment for Intelligent Network", pp 44-51, See pp 45-49 and Fig 4.	1-28